Programming architecture

an integrated algorithm design for a wooden building

master’s thesis 2018

Romain Fassotte

_ abstract

Tampere University of Technology,
School of Architecture

Master's Thesis 2018

Romain Fassotte

Examiner: Ilmari Lahdelma

In this thesis work I will argue for programming in the field of architecture. Through the realization of a wooden building – for the Vaasa's Wooden Tower of Innovation competition – I am going to explore and demonstrate how automation can be implemented in the design and production processes of a project.

The first chapter will briefly set the theoretical background of the thesis. It will define what is the approach behind programming and how it is being integrated in the architectural practice. The aim of the chapter is to introduce this new method of thinking by explaining the concept of parametric design and digital craftmanship.

The second and core part of the thesis will be an exploration of how automation can be integrated and serve in the design and production processes of a project. The task chosen for the design is the Vaasa's Wooden Tower of Innovation competition. The intention will be to show how algorithmic based design eases the realisation of complex geometries, from conception to manufacturing phases. Every step of the process will be explained in order to demonstrate where parametric design has been used and how it has been benefitting the process compared to traditional thinking.

Finally, the conclusion will summarize the ideas developed in the thesis by presenting a synthesis of the workflow and projecting how this new digital approach could change the way architects work in the coming years.

0_ glossary

_Algorithm: set of mathematical rules that, especially if given to a computer, will help to calculate an answer to a problem.[1]

_CAD, Computer Aided Design: design that utilizes computers to create and manipulate geometries in a digital environment. E.g. AutoCAD and Rhinoceros are CAD software.

_CAM, Computer Aided Manufacturing: technology that uses computer software to facilitate or automate a manufacturing process.[2] E.g. CNC routers and 3D printers are CAM machines.

_CNC, Computer Numerical Control: defines machines that use numerical coordinates to operate. Often used to mean CNC milling machines.

_Computation: use of computer to realize mathematical calculations or algorithms.

_Computational Design **(also** ***Parametric Design*****)**: design that utilizes computation as a method.

_Definition **(also** ***Script*****)*****:*** a set of rules that solves a problem or generates a solution. In computational design, the term is often synonymous with algorithm.

_FEM, Finite Element Method: numerical method for solving algebraic problems. Typically used in structural or fluid analysis to produce comprehensible visual models.

_Mesh: polyhedral object defined by a collection of polygons. It is a type of surface or geometry made exclusively of planar faces. Meshes are usually opposed to NURBS geometries.

_Minimal Surface (also Mesh Relaxation): surface that locally minimizes its area.[3] In computational design, minimal surfaces are obtained using Force Density Method.

_NURBS: non-uniform rational basis spline. NURBS-based geometries are generated with mathematically defined continuous curves. NURBS geometries are usually opposed to Meshes.

_Parameter: defined data, e.g. a geometry or a number, that sets a solution to a system, e.g. an algorithm.

_Utilization: in structural analysis, defines the ratio between the stress capabilities of a material and its stress under a certain load case.

***_Visual Scripting* (also *Visual Programming Language, VPL* or *Diagrammatic Programming*):** programming language that uses graphical components instead of textual code. Visual Scripting is often used in 3D modelling or web development for its user-friendly approach.

1: Cambridge Dictionnary, *Algorithm*, Cambridge University Press, 2018, https://dictionary.cambridge.org/fr/dictionnaire/anglais/algorithm, accessed 02.2018
2: Techopedia, *Computer-Aided Manufacturing (CAM)*, 2018, https://www.techopedia.com/definition/4698/computer-aided-manufacturing-cam, accessed 04.2018
3: Encyclopedia of Mathematics, *Minimal surface*, 2017, http://www.encyclopediaofmath.org/index.php?title=Minimal_surface&oldid=28248, accessed 04.2018

1_ Algorithm & Architecture

introduction

1_1_ definition

For several decades, the world has been touched by a wide technological shift that has deeply affected us on many levels. Algorithms are ruling the technology that we use daily, acting as the brain behind every digital mechanism we employ. They call the elevator when we push the button, they collect information to give us the weather forecast, they analyse the words we use in our web research, etc. As they are digital objects hidden behind screens and chips, they are extremely difficult to perceive although they are the key of any electronic system. As the notion may appear abstract for most people, it is pertinent to explain what defines an algorithm before getting into its impact on architecture.

An algorithm is "a set of mathematical instructions or rules that, especially if given to a computer, will help to calculate an answer to a problem" (Cambridge Dictionary). In other words, it is the artificial logic that has as main goal to solve a problem. As an example, in mathematics, we can compare a multiplication to an algorithm: the function *[3*(4-2)=]* gives us the result *[6]*. In this case, the calculus is the algorithm as it is the set of rules that provides a result, the rules being *x*(y-z)=u* . No matter what you set as x,y and z, the algorithm will provide u. In programming, coders are writing lines of algorithms to create mechanisms. It is what is used to design software and functions. When we type on our keyboard, an algorithm translates the physical action into letters that will appear on the screen. Later on, when we want to print those letters, another algorithm will interpret the file and move the printer that will provide a sheet of paper with the printed text. Those mechanisms, or technologies, are made possible by the simple fact of succeeding different rules. Once put together, they solve problems, generate actions, or whatever else the creator of the algorithm wanted to achieve.

When it comes to architecture, we can now understand how algorithms can serve in the practice. In fact, starting from the late twentieth century, architects have experienced the first digital revolution with the rise and development of Computer Aided Design. As in most of the professional fields – industrial, financial, medical, etc. - of that time, computers came as a new revolutionary tool to work easier and faster. It widely opened the scope of possibilities and definitively changed the way we worked, especially in creative disciplines like architecture. With the help of algorithms, engineers and programmers created interfaces that recreate 2D and 3D environments where designers could digitally materialize their ideas. Shapes and objects could be designed in a mathematically defined world. It means that instead of drawing with a limited scale and precision like on paper, geometries could be drawn without scale and with an unlimited level of precision. For architects, CAD software offered new possibilities of design. They could create shapes that were directly viewable and editable in 3D, which had not been possible with the traditional descriptive geometry or model making. Thanks to algorithms, complex geometries are now defined by their mathematical functions and not by their Euclidean projections. Every kind of geometry, from a simple line to a complex topology surface, can be virtually modelized with an unlimited level of detail. This had as impact that those geometries, once defined, could be used for concrete applications such as in architecture. Designers could digitally create a freeform volume and transform it into a building by adding a structure, walls, etc. Besides helping them in their productivity, the bigger change has come from the fact that CAD programs opened a whole new world of design possibilities for architects.

Nevertheless, even if those emerging technologies revolutionized the way designers were producing their designs, it didn't change the way they were used to think. The method was similar to the process that was utilized previously as the designer still used drawings to materialize their ideas, like with a pen and a sheet of paper but with an improved tool. Algorithms became more and more present in the practice but stayed as "behind the scene" elements. They progressively settled by being integrated in the design process but as a defined design tool, they remained with their own constraints or scope of possibilities. Indeed, like any tool, CAD software have their own design language. They can be used only within certain limits and are not universal. Whatever new possibilities the tool can have, it will still behave by following the rules for which it's made for. The designers had the choice to use the appropriate software for their needs but they couldn't design their own tool.

What has really changed in the past decades derives from the fact that designers could not only use CAD software as predefined tools, but they could also create their own technologies by developing algorithms themselves. The technologies that were previously closed and imperceptible now became accessible. Indeed, for a long time programming has been only open to informaticians or programmers. A designer couldn't create his own algorithm without having strong programming knowledge. This has changed as nowadays programming methods have become more and more user friendly, letting everyone access the topic. Once they could create their own tools or technologies, the scope of possibilities became much wider. Algorithms became used for various purposes and started to be integrated in every step of the design process, from the generation of complex geometries to the automation of the production.

It is in this context that we will speak about algorithms and architecture. As computational tools have now been made accessible to designers, it is important to embrace the subject and see how this emerging practice is being implemented.

1_2_ parametric design

As computational tools became increasingly implemented in architecture, it also impacted the objects that were produced by such methods. Algorithms brought new possibilities in terms of design and geometry. It opened a new world of complexity that architects challenged to design and produce. When we look at the architecture created by some of the pioneers of parametric design such as Zaha Hadid or Wolf D. Prix (Coop Himmelblau), it is obvious to say that they created a complexity that highly differs from the common built environment. This new technique has permitted a creation of objects that completely contrasts with the shapes we were used to seeing. This new aesthetic based on rules rather than drawings lead to an architecture that has never been experienced and that totally innovate and widen the design possibilities of the built environment. As being a behavioural revolution, parametric design has become a style under the name of Parametricism. In fact, with his issue of the 'Architectural Design' magazine, Patrick Schumacher presents Parametricism as the new style of the 21st century. With a delicate thought, he claims: '*Parametricism is architecture's answer to contemporary, computationally empowered civilisation, and is the only architectural style that can take full advantage of the computational revolution that now drives all domains of society*'[4]. The claim can sound a bit straight forward, but it is yet what parametric design strives to be.

Even if this thesis work does not aim to belong nor answer to this specific style, it still embraces the practice by showing how architecture can be created in a parametric manner. Too often parametric design is perceived as a free complexity or *an esoteric design process fetishism*[5], but it is necessary to demonstrate the power of the method that goes further than the aesthetic. As introduced above, algorithm design is a method that can be used in many more domains than only in the making of complex geometries. It's a way of thinking that can be integrated in the whole design process to ease and elevate the designers work. Although nowadays its main use occurs when realizing complex shapes, the principal character of such technique resides in its ability to adapt to precise project variables. Either those parameters are structural, aesthetical or functional, they generate the project by giving a unique solution of the parametric model. This implies that the project answers precisely the goals it has been designed for and it is with those perspectives that we should understand parametric design as a powerful design method for architects.

4: Schumacher, Patrick, *Parametricism 2.0 : Rethinking architecture's agenda for the 21stcentury*, London: AD Architectural Design, vol. 86, 2016, p. 10.
5: ibid.

1_3_ digital craftmanship

In addition to be a strong design tool, algorithms have also the strength to be a powerful aid when it comes to production. As the shapes and buildings we produce become more and more complex, it is essential to consider how to physically materialize the designs. The freedom we acquire when designing with 3D software leads to new forms of production that require specific methods and skills. Designers must develop task-specific production workflows that have to be fully integrated in the process. This make them more conscious towards the objects they are creating as it requires them to find detailed solutions adapted for the design.

In his book 'The Alphabet and the Algorithm', Mario Carpo explains how this new phenomenon is leading architects towards the previous understanding of the architect as a craftsman.[6] Indeed, before the Renaissance, architects were considered artisans as they were creating and building without instructionals drawings. The architect was supervising the project in person like an artisan. From the advent of *Alberti's authorship*[7], architects started to work as master designers transmitting the design instructions to the actual builders. The architect was the author of the idea, but the builders were responsible to materialize the design. This means that the architects started to use drawings as intermediate medium to transmit their instructions to the artisans: the builders. From that period and until today, designers entered the *notational regime*[8], using an intermediate language to realize their ideas. Euclidian projections became the common medium of production and communication and it strongly impacted the profession. As planners, architects took a certain distance from the physical production and lost their role of artisan.

This gap between design and production is about to disappear with the rise of digital manufacturing. New technologies such as CNC or robot arms allow to create digitally designed custom elements. Contrarily to traditional building methods, this type of fabrication is directly linking the virtual design to the physical object. The development of those new types of machines aims to ease the fabrication of custom objects and therefore bridges the gap between standardization and customization. Complex shape can now be realized by integrating specific details in the design process. Designers must develop their own custom solutions to realize their ideas and that reconnects them to a role of craftsman.

6: Carpo, Mario, *The alphabet and the algorithm*, Cambridge: The MIT Press, 2011.

7: ibid., p. 20-26.

8: ibid., p. 28-35.

1_4_ Rhinoceros and Grasshopper

In this thesis, all the algorithms will be developed with visual scripting through the use of the Rhinoceros software and its parametric design plug-in Grasshopper. Visual scripting is a type of programming that allows the creation of scripts in a user-friendly way. Instead of actual coding, definitions are made with predefined components that the user plugs together to create a desired function. This method has made programming accessible to designers without the need of specific programming skills.

The most common software used for this method is Rhinoceros combined with Grasshopper. (fig. 1) Rhinoceros is a powerful 3D modelling software that uses NURBS and mesh geometries. It has been initially developed for jewellery and product design but has been extensively used in architecture due to its strong capabilities in manipulating complex geometries. The software itself does not offer parametric design possibilities but needs a specific plug-in called Grasshopper to add those functionalities. Grasshopper is an interface that allows algorithmic design within Rhinoceros. It offers a wide range of components that generate and control geometries in the Rhinoceros interface.

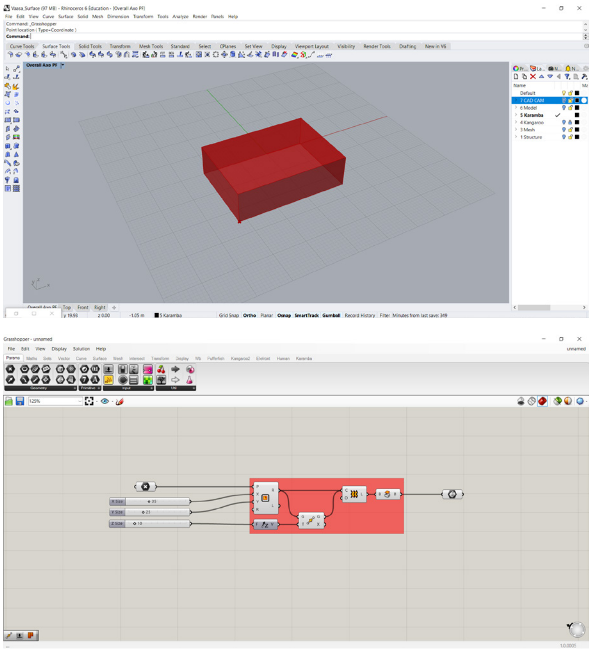


Fig. 1 Rhinoceros (top) and Grasshopper (bottom). Example of a box: the box is generated by using components in Grasshopper. Its geometry is visualized in the Rhinoceros interface. If a parameter (x,y or z) is adjusted in Gh., the geometry is simultaneously modified in Rhino.

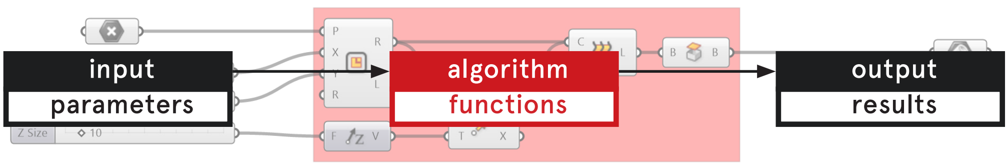


Fig. 2 Scheme of an algorithm made in Grasshopper with visual scripting.

2_ Algorithmic design for a wooden building

design

2_1_ Vaasa's Wooden Tower of Innovation

To explore the capabilities of computational tools in the design process of a project, I decided to take part into the competition for the Vaasa's Wooden Tower of Innovation. The competition took place in Vaasa, Finland, during the winter 2018 and was organized by the Tampere University of Technology's School of Architecture in collaboration with Koy Bock's Corner Village and the City of Vaasa. The purpose of the project was to design a new high-rise building in the northern part of Vaasa that would constitute a hub for innovation and act as a landmark for the city's entrance. Located next to the Bock's brewery, the building aimed to be the major element of the creative complex.

The design task consisted of a multi-faceted program that includes:

- flexible workspaces to be rented by different companies or organizations (1400 sqm);
- an auditorium for 350 people (360 sqm);
- a laboratory space conceived as a large FabLab for researchers and product development (340 sqm);
- lodging rooms to host researchers or other workers for short-term stays (300 sqm);
- a cafeteria (150 sqm);
- a lobby (100 sqm);
- maintenance and service spaces.

The goal of the competition was to propose an concept that would complete the development of the Koy Bock's Croner block and provide a village-like atmosphere for the whole complex. According to Jura Mikkonen, the Project Leader of the Koy Bock's company, the building had to "represent the Silicon Valley of Finland" and be "a cathedral for the whole Village". Their idea for the area was to create a place where people from a wide range of profession could come to work and stay the whole day. The goal was to provide all the necessary services to let users spend the entire day without having to leave. The place should contain different buildings and spaces that would allow economic, social and cultural activities. All-in-all, the complex should provide a warm atmosphere and be open for the public. As a new prominent place in the periphery of Vaasa, it should be attractive for the city's inhabitants and welcoming for events such as markets and concerts.

The site is a mainly residential area, the surroundings of which are mainly wooden villas. Some industries and warehouses are situated in the northern side and couple of commercial buildings in the western side of the site, next to the main street, Gerbyntie. The competition area is currently used as a parking and is the last part of the Koy Bock's Corner block that is presently unbuilt. A small wooden house still in use is located in the southern corner of the village and an unbuilt green area must be preserved at the back of the plot. There are two current vehicle access points, one on Gerbyntie and one on Pohjolankatu. Finally, a small river ows on the western corner of the plot and must be somehow integrated in the design. (g. 3 & 4)

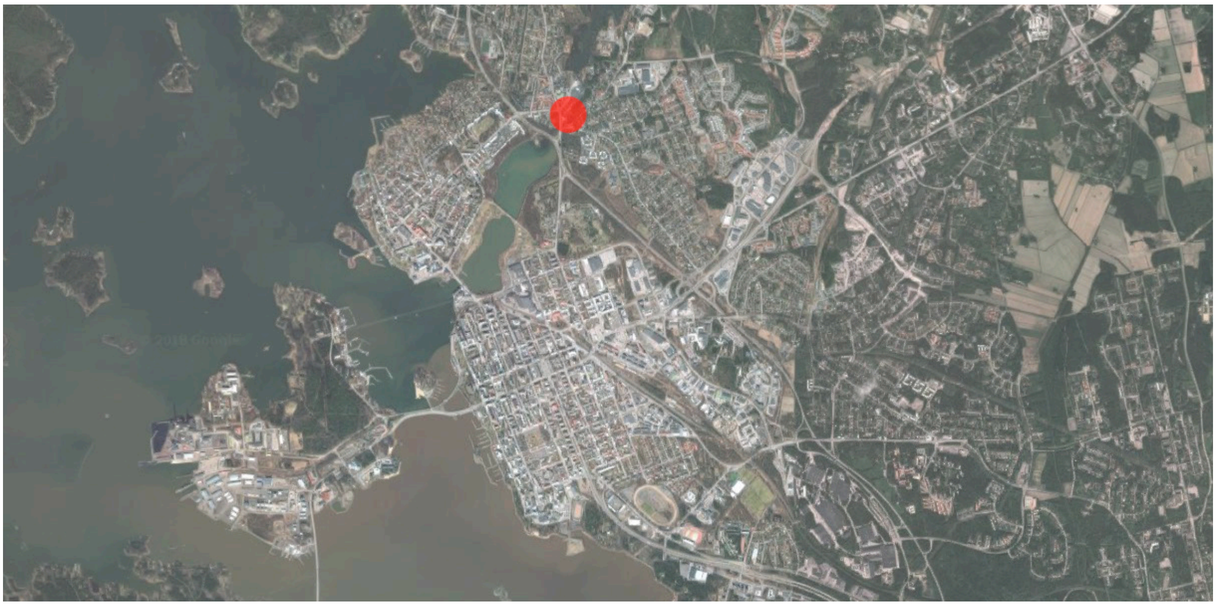


Fig. 3 Satellite view of Vaasa

Fig. 4 Site plan of the Koy Bock's Corner Village

Besides being in line with the innovative approach that concerns this thesis, the program was also interesting in the fact that it focuses on wooden construction. Indeed, another purpose of the competition was to highlight the use of wood as a building material by presenting a ground-breaking solution. As most of the contemporary large-scale constructions being parametrically designed are built using steel or concrete, it is a real opportunity – and challenge – to propose a solution made entirely with a sustainable materiel like wood. In fact, when we look at the actual tendencies in the building industry using algorithmic aided designs, we can observe that most of the large freeform buildings are made with complex structures using an abundance of steel and concrete elements. It is unfortunate to see that too often interesting shapes are wasting a massive amount of material to make them possible, like for the Guggenheim Museum of Bilbao or the Heidar Aliyev Centre to mention iconic examples. (fig. 5 & 6) Therefore, having the opportunity to explore complex geometries with a locally abundant and sustainable material was a real interest.

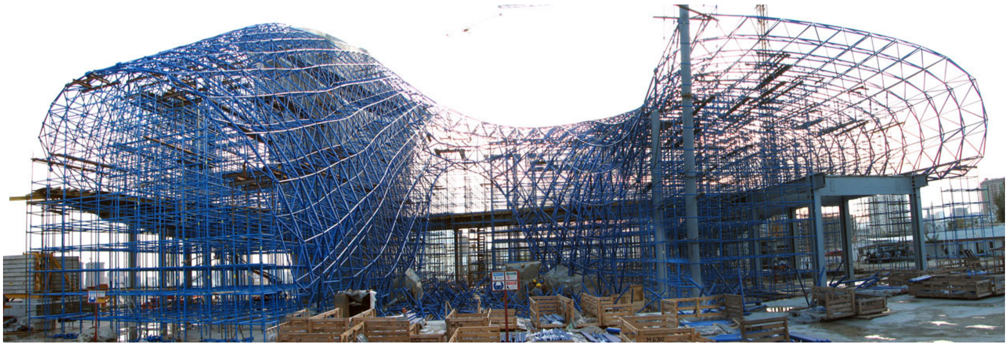


Fig. 5 Steel structure of the Heidar Alyev Centre

Fig. 6 Construction of the Guggenheim Museum of Bilbao

Note: I would like to point the fact that both the program and the site where defined by the competition organizers and are therefore not part of the thesis. Even if the purpose of the competition can be questioned, it does not take part in the thesis' subject.

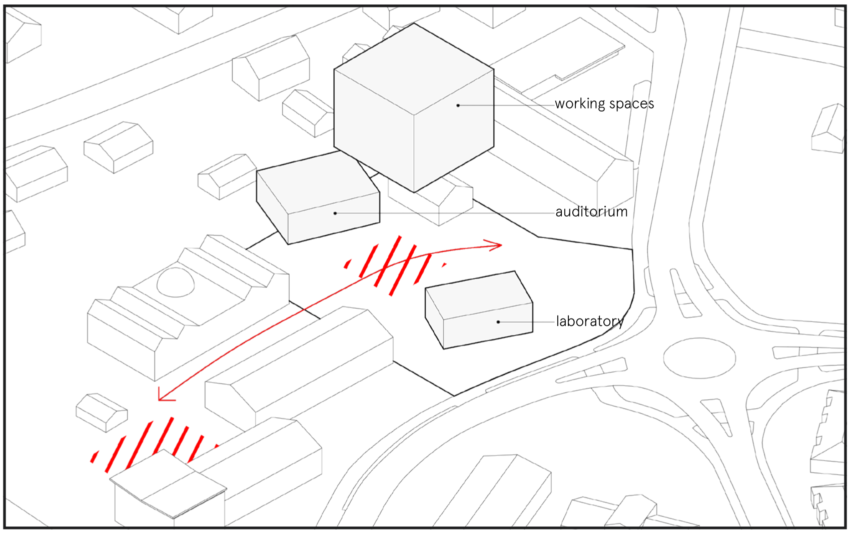
2_2_ proposal

In accordance with the competition organizers vision of proposing a village made up of diverse spaces and buildings, the idea was to extend the existing structure and reinterpret the concept of village by creating a building gathering different entities together around a common public space. The design would contain the main functions as readable volumes and host different public spaces and paths. The idea consists of creating a building like an environment that can be explored and used in various manners. On top of being experiential, the project should also be evolutive in time and inspiring for the creative workers for whom it is made for.

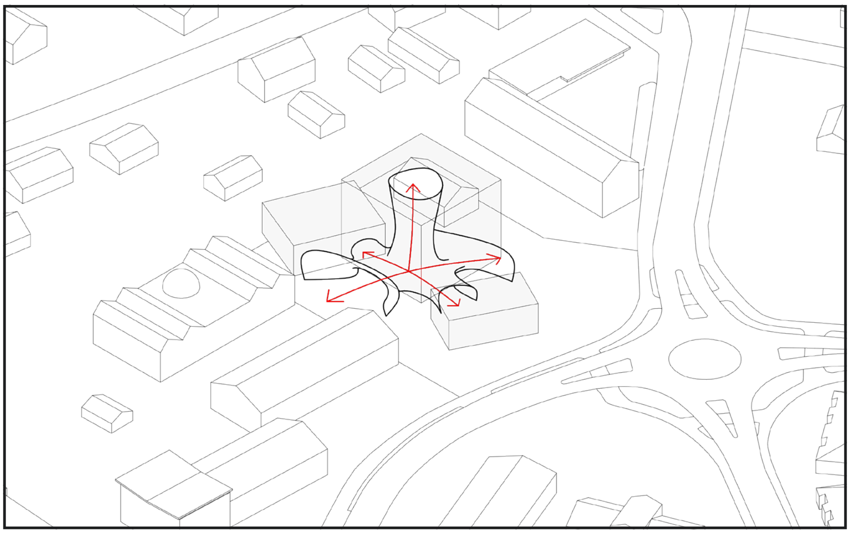
The approach consisted of having the three main functions, namely the working spaces, the laboratory and the auditorium, as singular volumes clearly identifiable from the outside. This would simplify the appreciation of the wide program and ease the users' orientation. The laboratory would be placed on the street side to showcase the workers' activity, the working spaces on the higher levels and the auditorium on the back of the plot, directly linked to the entrance.

To connect and organize the three masses the idea consisted of creating a three-dimensional shape that would act as an extension of the existing village's street. Placed in the centre of the building, the freeform space would act as a connecting interface between the different functions and foster the interactions between visitors and users. It would integrate two inviting entrances on the sides and offer a sufficient area to be used for different purposes such as exhibitions or meetings.

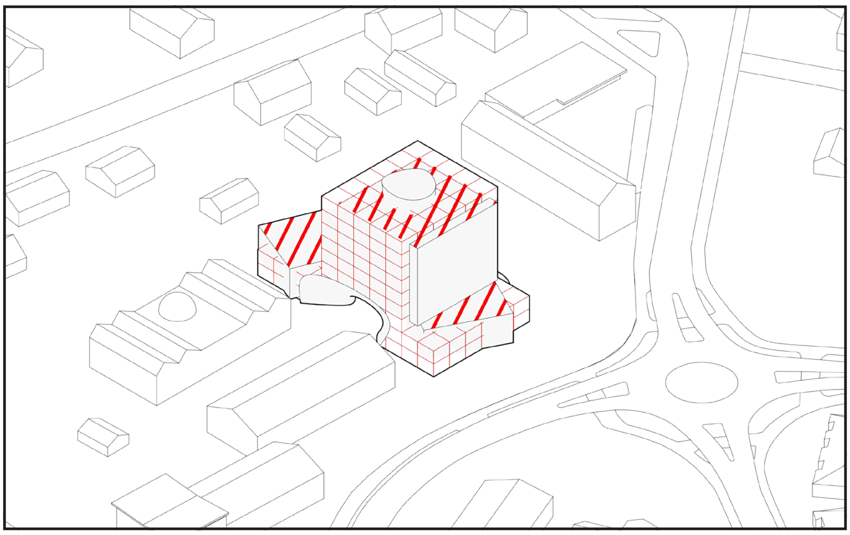
On the outside, the goal consisted of extending the inner structure to provide both a wholeness to the building and a support for future development. The grid could be used to create new spaces or functions, making the building evolutive and future-proof. As covering the whole edifice, it would be roamed by a small path giving the access to the different piazzas created by the roofs.



extend the street within the building



connect the functions through a freeform space



broaden the structure and provide piazzas

Fig. 7 Sketches of the proposal

2_3_ design phases

As always in an architectural project, the design starts with an idea and then evolves through the design process. The designer tries to follow their design objectives while integrating the constraints and parameters deriving from the task. This process often implies considerable changes and demands a lot of effort by designer who has to successively adapt their design.

When we work with algorithms, the concept can be considered as a constant. It is the direction in which the designer tries to head towards. It is the set of rules they set to themselves to create the final object. For example, if the goal is to create a cylinder, the designer may not know its length or radius but they can set the rules that generate the cylinder. No matter what the input values will be, the design will remain as a cylinder.

This chapter will go through the design phases of the building by explaining how algorithm have been used to translate the abstract idea into the final project. I will start by demonstrating how I used computation to generate and shape the different elements of the project and then I will explain how the structure has been developed and analysed.

2_3_1_ formfinding

2_3_1_1_ scale

The first step that commonly emerges when creating a project is to define the scale of the building by analysing volumes of different sizes and positions. This phase is usually translated by the production of modules placed in a model, either physically or digitally. In that situation, algorithms can be used to mass produce various iterations of the modules that can be modified within predefined parameters. By defining few rules and selecting some parameters on which to act, the algorithm can automatically generate a considerable amount of volumes' variations. Additionally to providing a first appreciation of the volumetry, the definition can be created to supply preliminary data like the gross area, volume, or the shadow extent.

In our case, the idea for the general shape of the building was already clear. It would have a large podium adapted to the height of the surroundings and a smaller tower somewhere in the middle. The volumes could be generated by two boxes placed on top of each other. Their height would be fixed as the amount and height of floors would be known. Two flexible parameters were defined to generate the volume: the length of the podium and the size (length + width) of the tower. Both parameters could vary between specific limits and generate a unique solution.

As well as generating the volumes, the script could also calculate usefull information from the geometry. To get a clear idea of the scale of the building, it was decided to output the total gross area of the building as well as the areas of both the podium and tower floors. Once defined, the algorithm could produce an unlimited amount of building's iterations within the defined limits of length and width. This could quickly and effortlessly provide a scale reference that can be used to start the design. Whenever the dimensions of either the podium or the tower would change, the definition would show if the building is still in the required scale. (fig. 8)

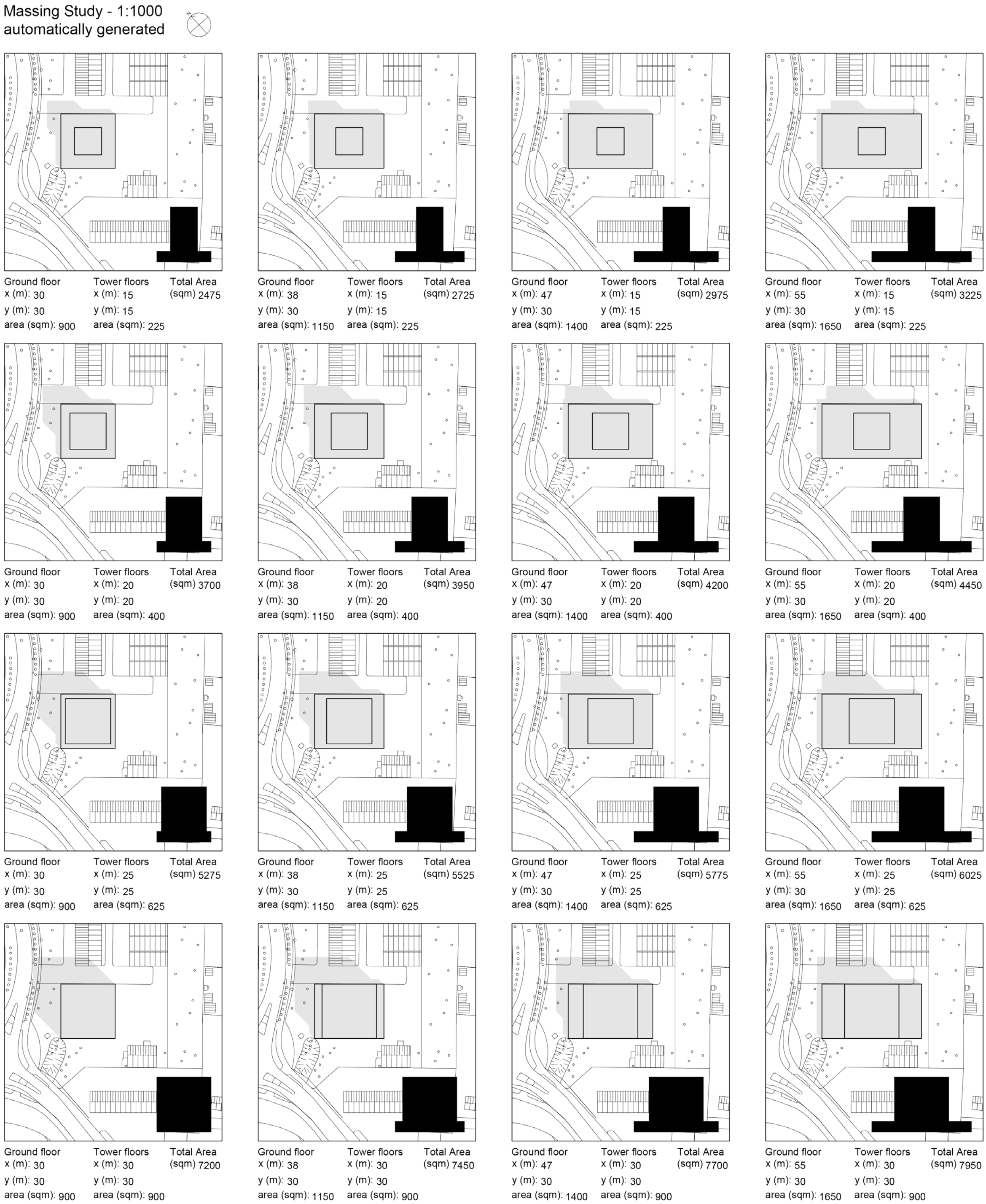


Fig. 8 Iterations of the preliminary volume. Automatic generation of the volumes, gross areas and shadow extent. Podium length varies in x axis, tower size varies in y axis.

2_3_1_2_ surface

The next and main phase in the design process consisted of generating the surface that would connect the three main functions and host the central public space. As the goal was to get a seamless flow between the different masses and openings, the idea consisted of creating a continuous surface linking every element. The surface must be articulated around two axes – the street and the atrium – and link three volumes – the working spaces, the laboratory and the auditorium. The resulting surface would resemble a pipe exploded on its base in every direction.

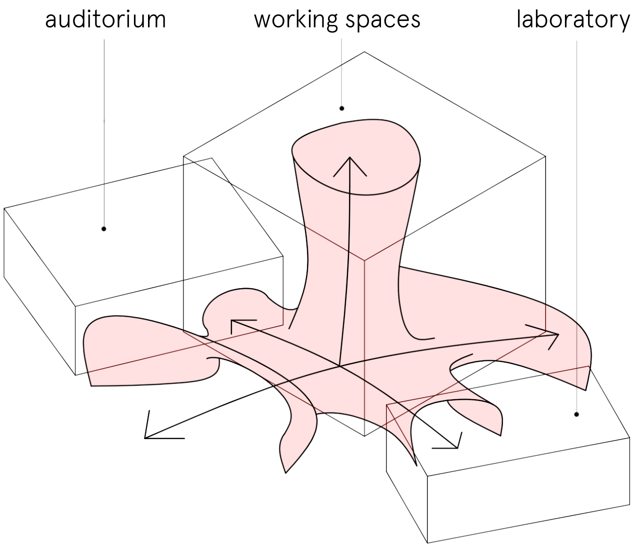


Fig. 8 Connecting surface.

As the typology of the surface is quite complex, the strategy to get a satisfactory result was to use the method of minimal surface. This method consists of generating a 'stretched' surface from a polygonal mesh. Geometrically speaking, the surface is calculated by locally minimizing its area.[9] This is accomplished by applying certain factors to reduce the length of the surface's edges or ribs. In some ways, it consists of replacing the surface's fibre by springs. Physically, it's like having a piece of fabric, hold some of its corners or edges in place, and add elasticity to it. The more elasticity added, the more the surface will stretch towards its centre. (fig. 9)

This technique has been made possible with the use of computational tools. Exactly like explained, the script is replacing the surface's wireframe by springs of a certain strength. The surface is then stretching itself as much as we increase the strength of the springs. The benefits of working with minimal surfaces derive from the facts that it erases the sharp angles of its original shape and that it can be made with any typology of surface. As it works using meshes and not NURBS, the shape can be generated no matter it's complexity.

9: Encyclopedia of Mathematics, *Minimal surface*, 2017, http://www.encyclopediaofmath.org/index.php?title=Minimal_surface&oldid=28248, accessed 04.2018

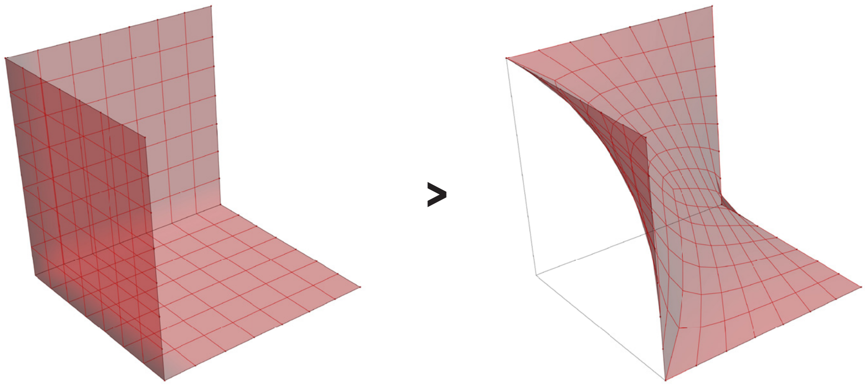


Fig. 9 Example of minimal surface from three planes.
Original surface (left) & generated surface (right)

When modelling freeform shapes, it is essential to understand how the geometry is made and how it behaves. Rational shapes are easy to understand as they follow simple rules and logic: a beam and a column follow an axis, a rectangular slab is characterized by two dimensions, a box is made with six rectangles, and so on. As they are easy to read, those forms can easily be implemented in a project and be used as structure or architectural elements. When it comes to freeform shapes, the logic is often unapparent at the first sight and it is necessary to understand and optimize the way the object is built in order to ease its development in the future phases of the project. In this case, the shape would serve as a three-dimensional wall and would therefore include structure and panelling elements. To get the most of flexibility, it was crucial to include enough parameters and control points in both the original geometry and the script. One of the biggest concerns encountered when creating an algorithm is the selection of the right parameters and defining them as flexible or fixed. In this case, as the surface is made with polygons, the adaptability of the shape would be defined by the position and the amount of control point the polygons would have. The idea was to create the shape by intersecting two pipes. One going from entrance to entrance and one from the top to the bottom. The control points were placed on the extremities, on the entrance doors and at the junction of the two pipes. (fig. 10)

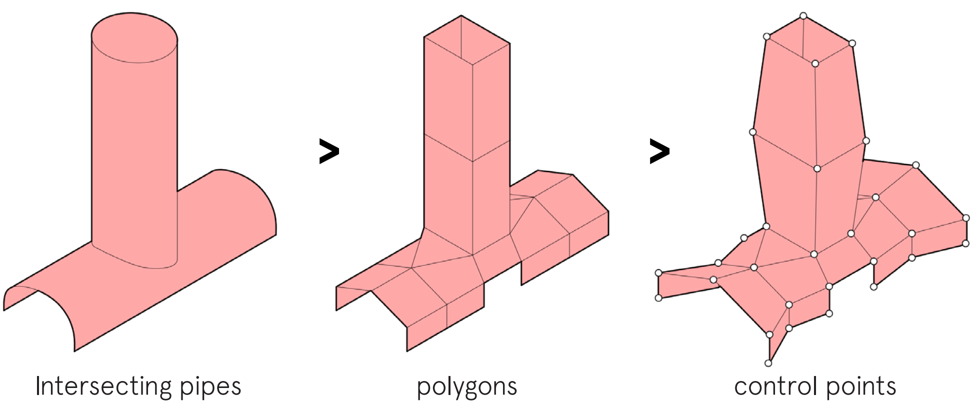


Fig. 10 Construction of the polygonal surface.

Other essential elements of minimal surfaces are their anchors. When the surface stretches, it must be attached to some points or curves. In this situation, the shape had to stick to the ground and to the openings – entrances and roof.

When all the elements of the shape have been modelled, the script can be created to generate the minimal surface. It is made using Kangaroo, a plug-in for Grasshopper that brings physical properties into the digital model.[10] In this case it is used to compute the tension applied in the surface. The algorithm acts with three main steps. It first divides the polygons into smaller ones to get a smooth mesh, then it identifies the points to which the surface has to stick and finally it computes the new surface by applying a predefined tension on it. The result is a smooth shape that seamlessly connects the different volumes and openings. (fig. 11)

Once the whole definition is set, the model can be transformed and adapted infinitely without the need for redesigning the complex shape, that being a major benefit of using parametric design. As the logic of the design is written with rules, the project can freely evolve while the geometry automatically adapts itself to the new settings. (fig. 12,13,14 & 15) This way of thinking allows the designer to get into the design of the plans of the different volumes without worrying about how much effort and time it would cost to redesign the 3D shape. With that method, a complex three-dimensional element can be integrated in the building with extreme fluency. An element that would have been exceedingly demanding to design and manipulate with traditional design methods has been turned into a nearly effortless process.

The complexity of the object still resides in the script itself but as it is conceived with the rules generating the shape, the benefit is that it gives to the designer a total understanding of the geometry they are designing.

10: *Kangaroo Phisics*, created by Daniel Piker, v. 2.42, http://www.food4rhino.com/app/kangaroo-physics

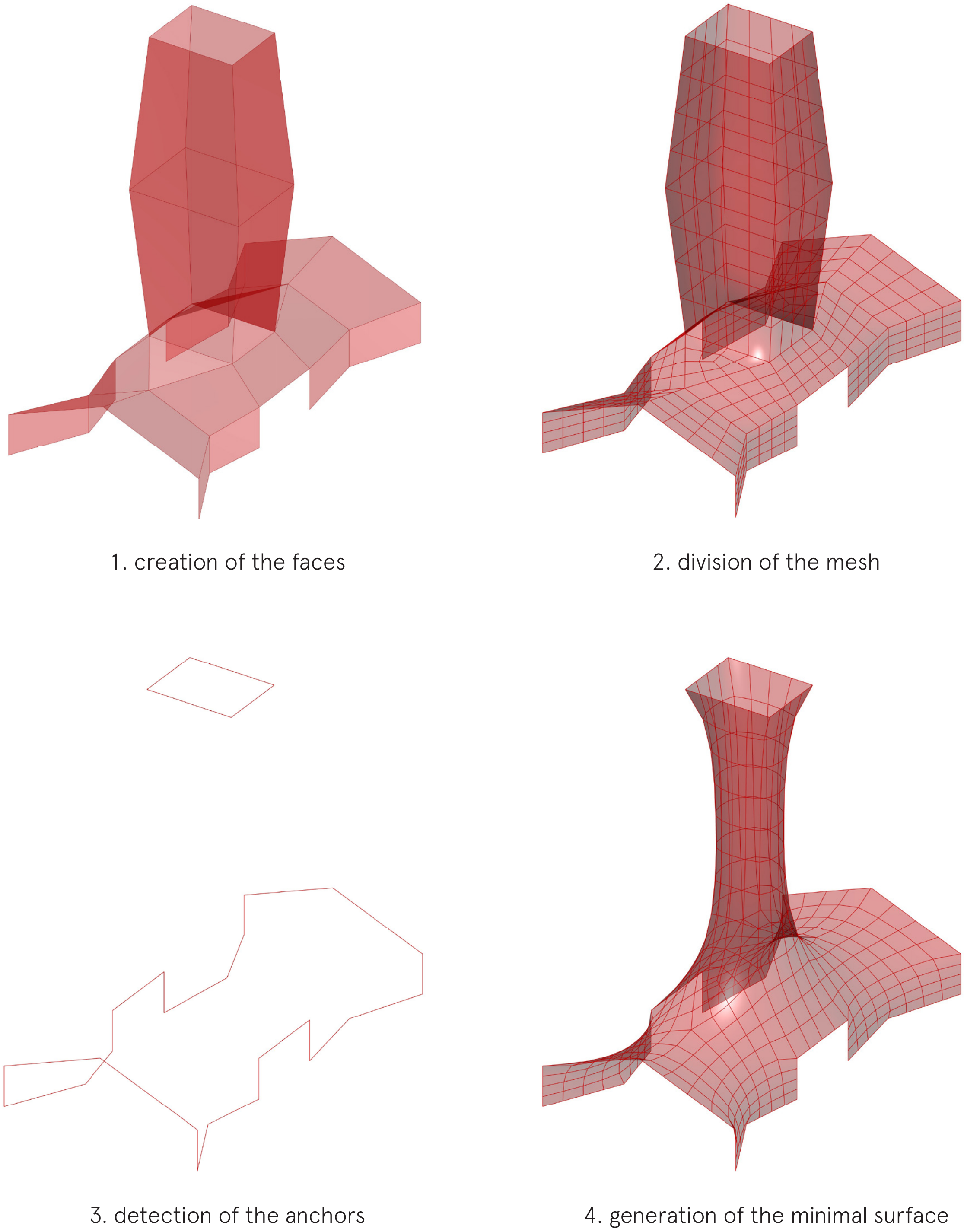


Fig. 11 Generation of the surface with Kangaroo.

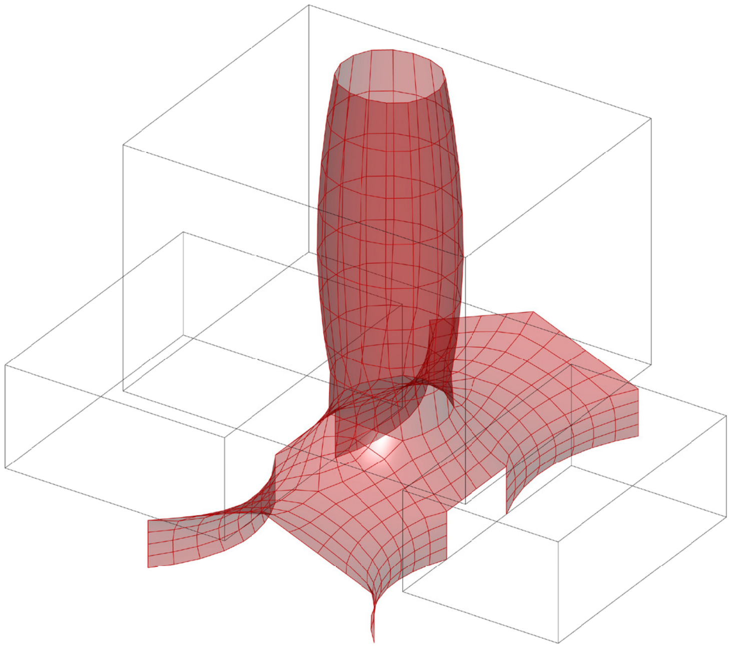


Fig. 12 Surface iteration 1.
Initial setting of the volumes and refining of the atrium

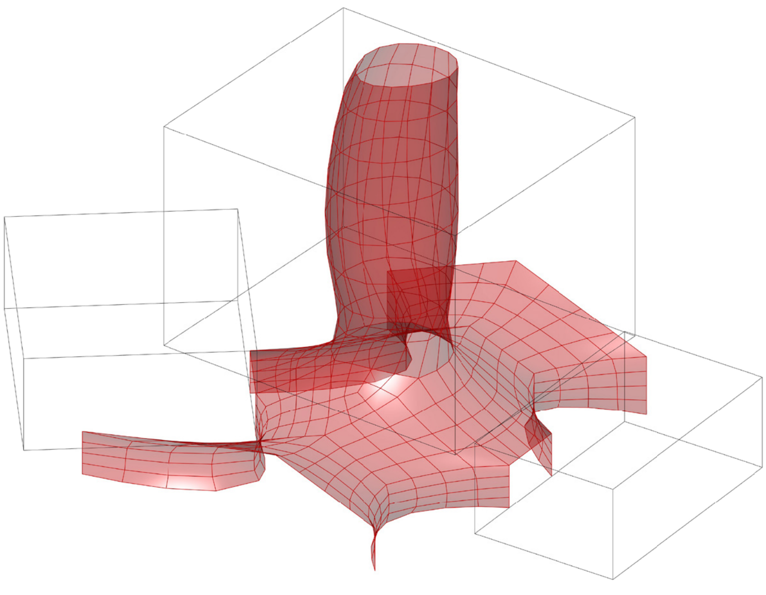


Fig. 14 Surface iteration 3.
Modification of the volumes' position and size.

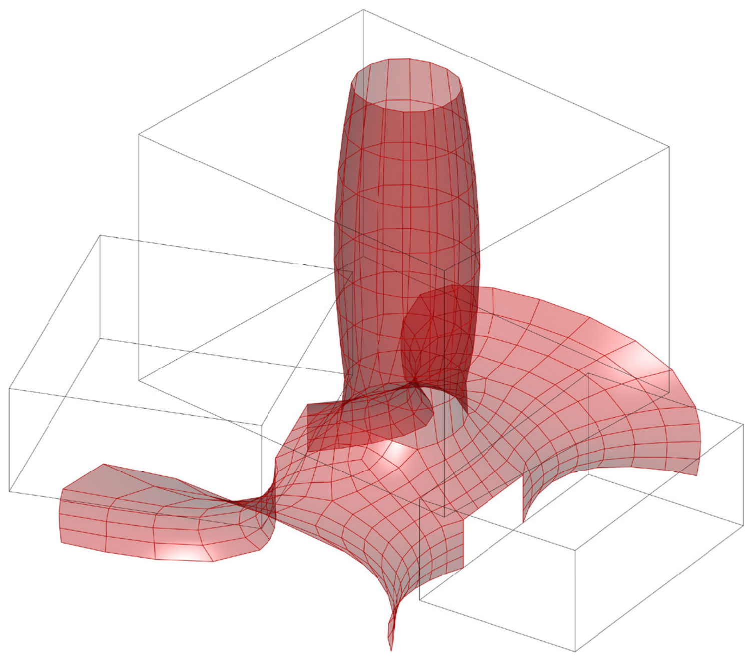


Fig. 13 Surface iteration 2.
Test with curved anchors at entrances' edges.

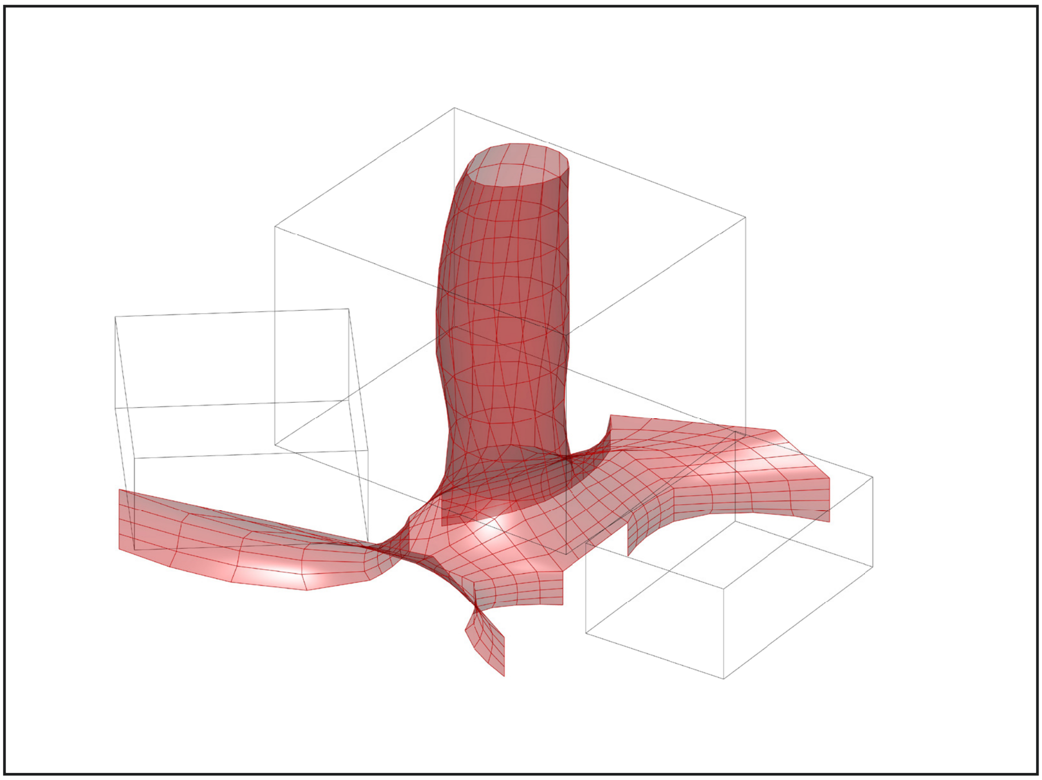


Fig. 15 Surface iteration N.
Final design.

2_3_1_3_ grid

The last element where computation has been used is the grid that defines the building. It would act as a structure for both the building and the paths linking the roofs. In architecture, designing a grid is always challenging as its dimensions are deeply impacting the configuration of the design. It provides a framework of the whole building and it is often difficult to know what measures would perfectly fit the design on its early phases.

Nevertheless, as a grid is a component that is made by simple elements repeating themselves under certain rules, it can efficiently be created with algorithms. The benefit of working parametrically comes from the fact that it allows the designer to adapt the grid during the entire project development phase without the need of repeating the same process after every alteration to the design.

In this case, the three-dimensional grid had to be done in two parts, both resembling the volumes made during the scale part: one larger podium and one tower located in the centre. Both grids would have the same dimensions as the structure has to be continuous for the whole building. The script created aimed to generate both the grid geometry and the structural element. It included the parameters for the spacing in the x, y and z directions and the thickness and width of the structure – beams and columns. (fig. 16) At the beginning of the design the spacing was set to 6*6 meters but it quickly appeared that those dimensions were too big when doing the layout of the rooms. After few successive changes the final dimensions of the grid where set to 5*5meters with a height of 3,50 m and the section of the timbers elements was preliminary fixed to 50*50 cm. With the final settings, the grid consisted of 1481 structural elements that were automatically modified whenever needed in the process. (fig. 17 & 18)

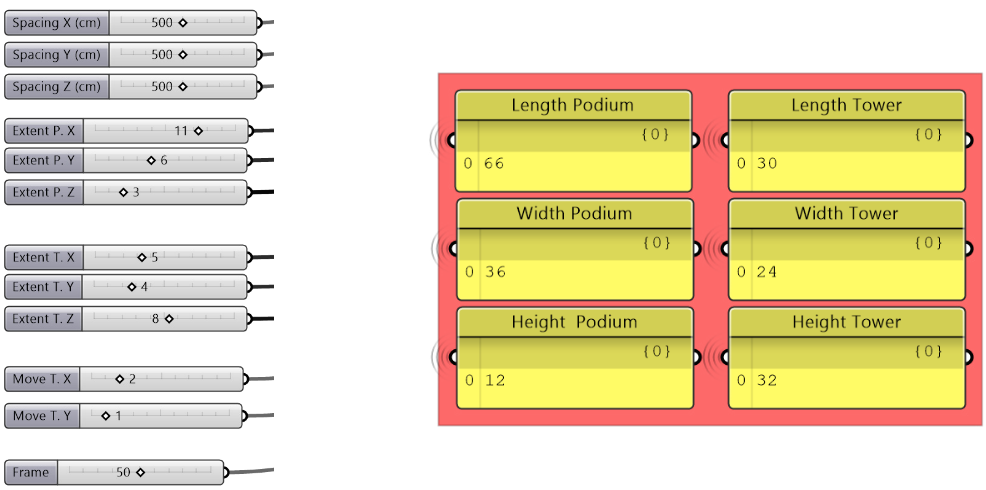


Fig. 16 Grid parameters and outputs.
On the left, the parameters such as the grid spacing or timber thickness allow to control the shape. On the right, the output data that instantly displays the dimensions of the structure (in meters).

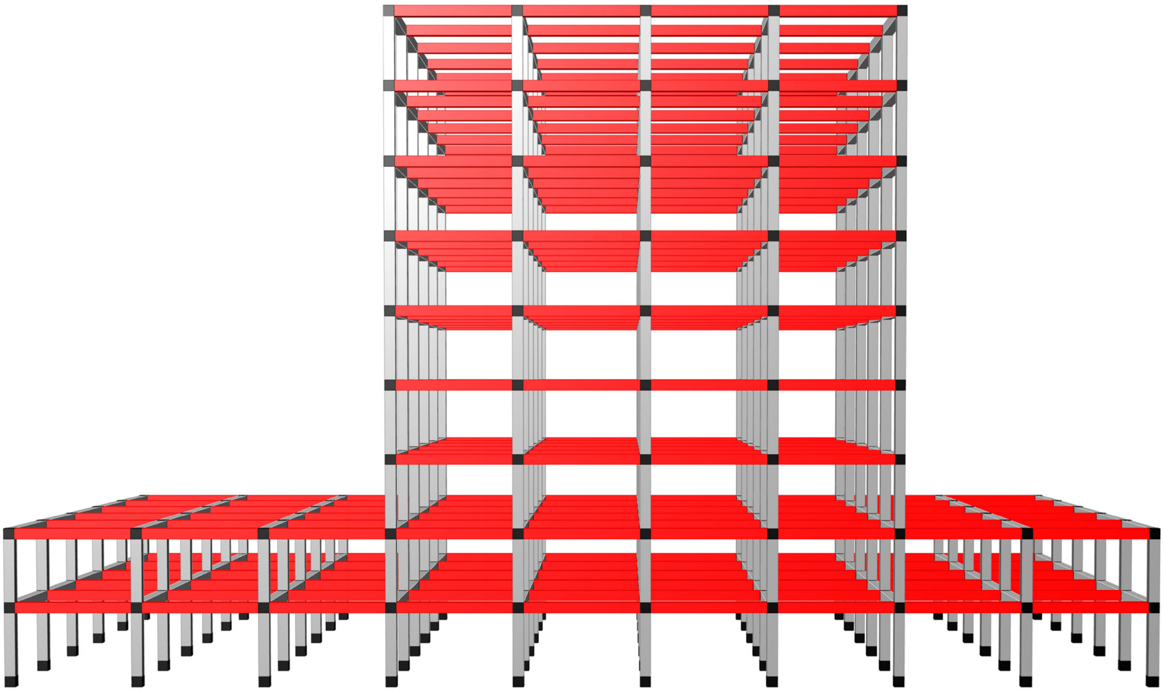


Fig. 17 Initial grid structure.
*Dimensions 6*6*3,5m. Section 50*50cm*

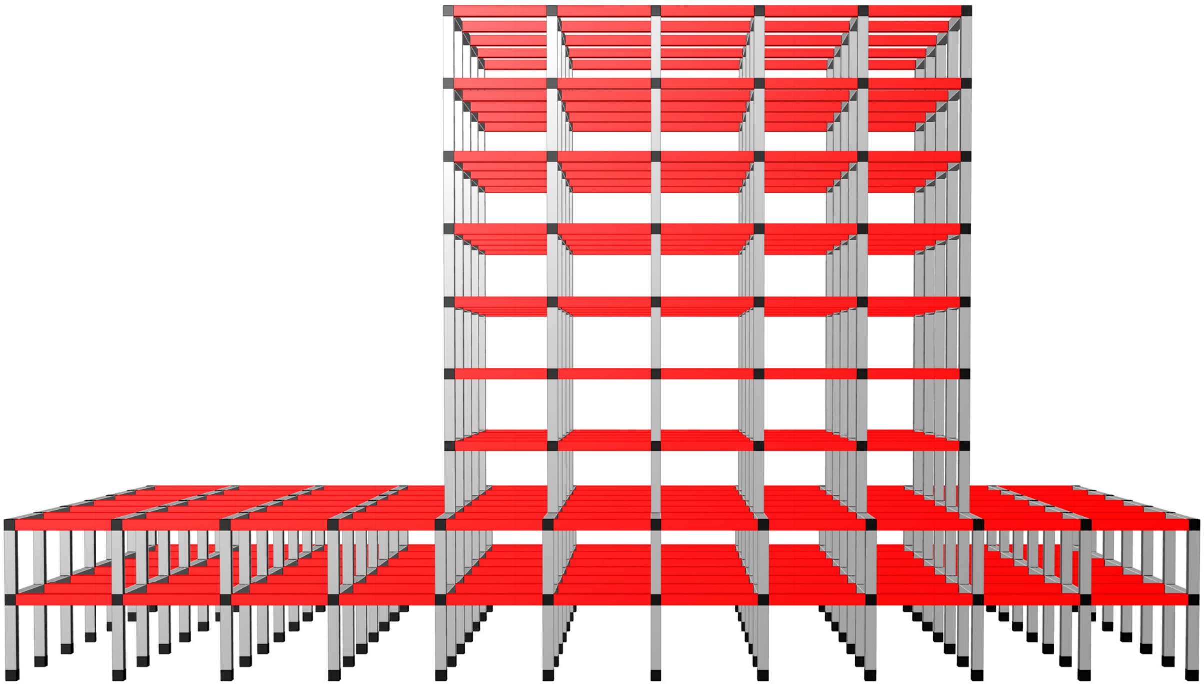


Fig. 18 Final grid structure.
*Dimensions 5*5*3,5m. Section 50*50cm*

2_3_2_ structure

The complexity that occurs in freeform geometries is often not following the common codes of construction and structure. When a building is designed with simple elements like straight vertical walls and rectangular slabs, it is easy to project how the structure will work and what details are critical. The designer intuitively has an idea of the scale to apply to the structure and they do not need calculations to know if the dimensions they are drawing are reasonable or not. However, when the structure flows in every direction and doesn't follow any rational rules, it is much more difficult to estimate its integrity. The manipulation of such structure becomes challenging as its arduous to get a clear picture of the entire object. This chapter will explain how the structure of the freeform surface has been designed and what the benefits have been made of using computational design.

2_3_2_1_ structural system

One of the initial steps when designing an architectural object, is to determine the structural system to be applied. This structural strategy is always defined by the architect according to the design's features: a parking garage would be easily made from a concrete posts and beams structure and a small cottage would likely be done with wooden logs. This decision is made during the design process of the project and will impact the design together with its evolution. When it comes to complex shapes, it is often difficult to apply a traditional structure with materials and forces flowing in three dimensions simultaneously. A custom structural system is then the most judicious solution.

Algorithms reveals to be decisive tools in these situations. In fact, structural systems are based on rules as they follow the laws of physics and material. They can be independently created and then applied on the shape to be realized. Once a script is made, the shape can freely evolve, and the structure is automatically modelled, calculated or analysed. This gives to the designer a seamless workflow between design and materialisation that eases the making of complex shapes.

In this case, two systems have been tested. As one part of the three-dimensional shape was apparent in the tower, it was essential to define a structure that would elegantly fit the design. The first idea consisted of creating a structure by slicing the shape. This type of structure allows to get a strong stiffness but is not especially effective regarding the use of material. Indeed, the structural utilization – the ratio between the material capabilities and its actual use – of the different pieces can be highly discontinuous. Some parts have a large area and therefore uses an extensive amount of wood but are not essential regarding the structural needs. Moreover, as this type of structure is defined by the plans, it has specific orientations that does not always fit the shape. This means that the different pieces follow straight axes that cut the shape without adjusting themselves to the face's direction. This leads to elements that are sometimes excessively wide as they are tangent to the shape and sometimes too thin when they are perpendicular to it. (fig. 19 & 20)

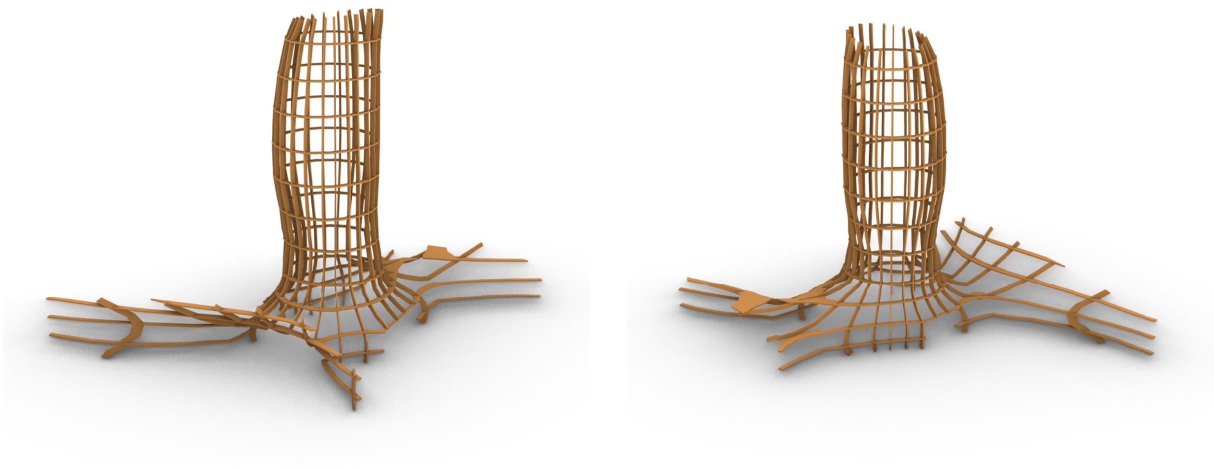


Fig. 19 Sliced structure with horizontal & rotating plans.
Substantial contrast between atrium and entrances structure.

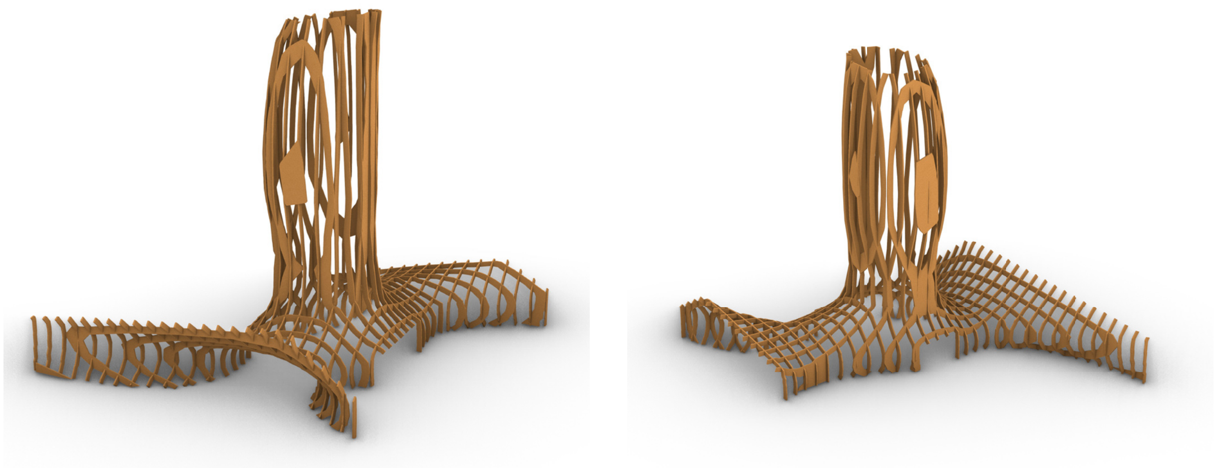


Fig. 20 Sliced structure with perpendicular plans (also called waffle structure).
Atrium's aspect not appropriate and elements too wide at some locations.

As the first type of structure wasn't optimized for the present shape, another system had to be used. The second type of structure consisted on creating a gridshell. Gridshells are structural meshes made with linear elements such as timbers or steel sections. They are generally created by applying or extracting a wireframe from a surface. As they can adapt to every directions, they are efficient system when applied to double-curved surfaces. Gridshells are usually formed with triangles or square-like shapes.

In this case the first step was transforming the shape into a triangle wireframe. As the goal was to cover the structure with wooden panels, it was necessary to get planar faces. The quad faces - square-like faces – that were obtained during the minimal surface generation (cf. chapter 2.3.1) were non-planar surfaces and would have been difficult to manufacture. (fig. 21) The goal was then to modify every quad into triangles to get planar faces. This would also increase the stiffness of the whole shape as it adds one local axis in the structure. A first transformation has been done with a built-in Grasshopper component but the result didn't create a continuity between the faces. The script had to be modified later to uniformly transform the quads and get a better continuity between the triangles' edges. (fig. 22)

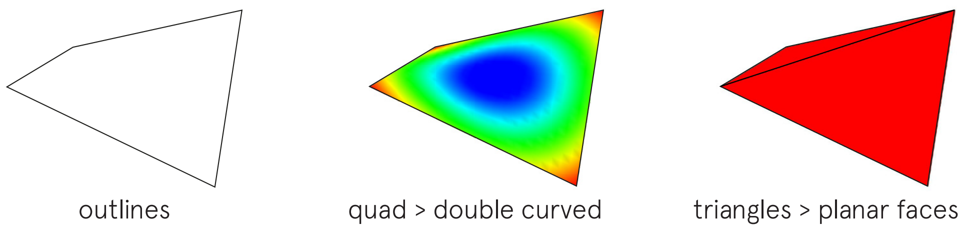


Fig. 21 Curvature analysis of a mesh face.
The quad face is double curved. If divided in two triangles, it becomes planar.

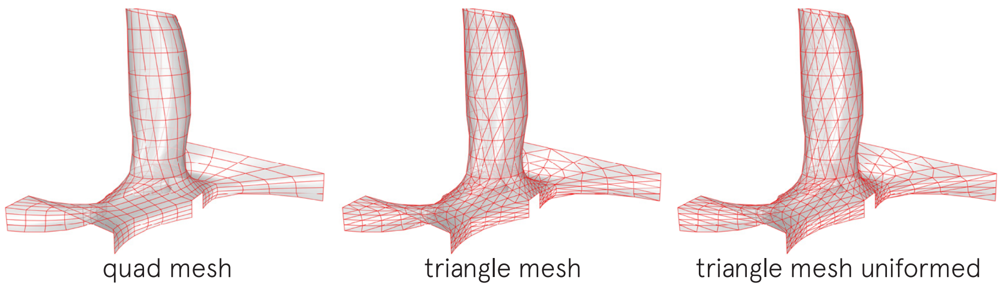


Fig. 22 Mesh modification from quads to triangles.

2_3_2_2_ structural analysis

Once the wireframe was created, it could be used to generate the structure and process a Finite Element Model (FEM) analysis. The software used for such evaluation was Karamba. Karamba is a structural engineering plug-in for Grasshopper that allows the creation of parametric FEM.[11] On top of being an actual calculation software for structures, it also provides components to easily visualize the stress and deformations occuring in the building. When linked to the architectural model, it allows to appreciate the efficiency of the structure in real time. The parametric model can be manipulated and modified with a simultaneous visualization of its structural performances.

The load bearing structure to be analysed consisted of the three-dimensional surface and the grid combined. The grid would be designed and act as the main structure and the surface would be the secondary element, designed to mainly bear its self-load. As wood is the main material to be used for the structure, the most appropriate wooden technology used for such construction is glued laminated timber. To simplify the calculation, a single type of glulam has been applied to the whole structure. The characteristic strength and stiffness values of the glulam elements have been defined using the *GL28h* class determined by the *Glued Laminated Timber Association*[12]:

Modulus of elasticity: $E_{0,k}$ = 12600 N/mm^2

Shear modulus: G_k = 780 N/mm^2

Bending strength: $f_{m,k}$ = 28 N/mm^2

Self-weight: ρ_k = 460 kg/m^3

In reality, the structure would be designed with rectangular beams linked together by steel connections (see chapter 2.4.1). Nevertheless, to ease the calculations of the structural model, the structure was dimensioned as entirely made of timber. It means that the structural performances of the nodes were parametrized as being the same as the ones applied in the timber elements. Even if this simplifies the structure, it did not affect the structural integrity of the building as the steel joints have a stronger resistance to bending compared to glulam elements.

The loads to be applied includes the live loads evaluated to 2 kN/m^2 [13] and the dead loads including (fig. 24):

Self-weight of the structure: $\rho_{s,k}$ = 460 kg/m^3

Wooden panels covering the surface[14]: $\rho_{p,k}$ = 440 kg/m^3 => ≈ 12 kg/m^2

Wooden slab including technical ceiling (estimation): $\rho_{f,k}$ = 150 kg/m^2

11: *Karamba,* © Karamba3d, v. 1.3.0, http://www.karamba3d.com/

12: Glued Laminated Timber Association, *Design data*, http://www.glulam.co.uk/about_designData.htm, accessed 03.2018.

13: *Eurocode 1: actions on building structures,* Moscow: EU-Russia cooperation on standardisation for construction, 2008.

14: value from the voluminal mass of the *Kerto® LVL L-panel* from Mestäwood.

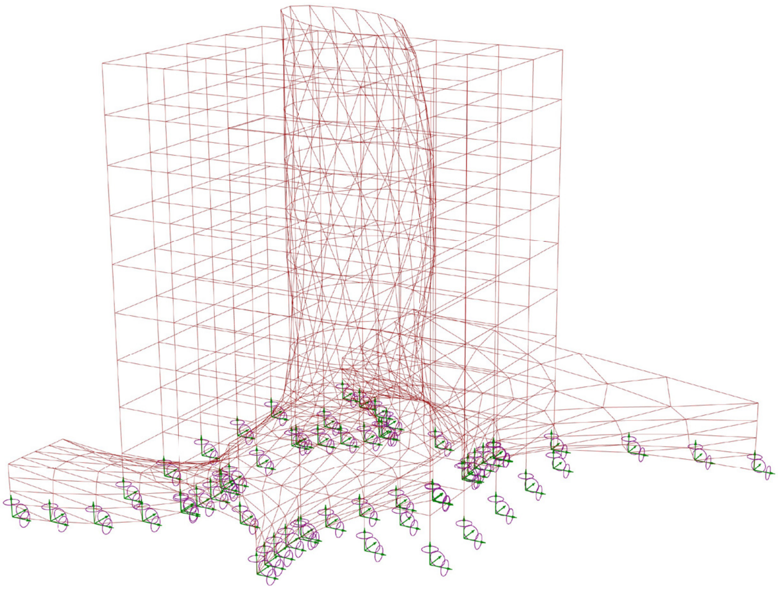


Fig. 23 Structural model: supports and elements.

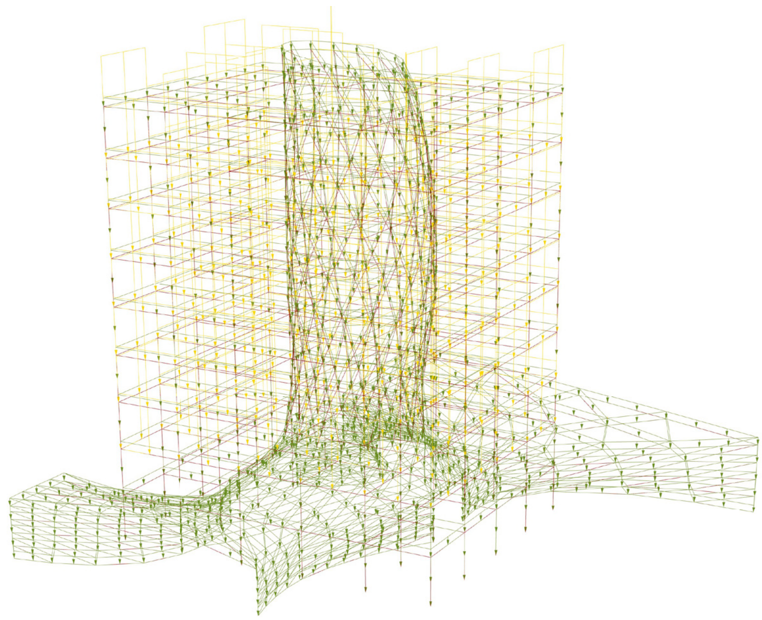


Fig. 24 Structural model: loads.

NOTE: It is obvious that for the actual construction of the building the structure would have been evaluated by including more loads cases and precision. Nevertheless, the goal of this preliminary analysis it to help the designer to optimize the shape and estimate the structure efficiency. In this way, the further phases of production have less chances to come with surprises and adaptations, cost and design wise.

The structural model made in Karamba provides a vast amount of data to understand the structural efficiency of the building. In this case the goal was to get a rough pre-dimensioning of the building. To do so, two types of values were analyzed. First the utilization of the structural elements, this gives the ratio between the structural capabilities of the material and its actual stress induced by the load case. In other words, if the utilization goes over 100%, it means that the structure collapses. The second value that must be looked at when dimensioning is the displacement. The displacement corresponds to the movement of a definite point of the structure from its initial state to its position when a load case is applied. It is an important feature to look at in a building as a structure can be self-sustained but have displacement compromising the use or the architectural appearance of the building.

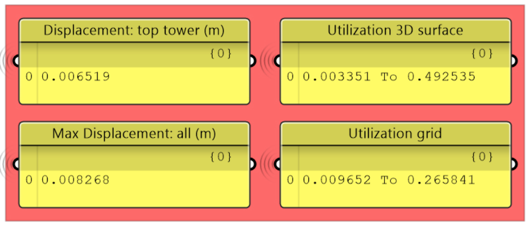


Fig. 25 Output data: displacement and utilization.
The maximum displacements is calculated for the whole structure and at the top of the tower. Utilizations is distincly evaluated for the grid and the 3D shape.

Once everything is set up, the FEM model becomes an easy tool to manipulate and evaluate the structure. The real-time visualization of the utilisation and displacement allows to do some fine-tuning of the different sections and quickly get an appropriate dimensioning. By showing the structure with contrasting colours, it also provides useful communication tool to share with clients or other stakeholders. (fig. 26 & 27)

Here, the structure was dimensioned regarding the displacement at the top of the tower and by trying to keep a maximum utilization of around 50% . Also, as the structure of the freeform shape was visible on its upper part, it had to match the design integrity. The idea consisted of having this part of the structure quite thin to maximize the views through the atrium. After few tests, the structure was defined with sections of 40*15cm for the 3D shape and square sections of 50*50 for the posts and beams of the grid. Like so, the maximum displacement at the top of the tower was 6mm and the utilization didn’t exceed 50%. (fig. 25)

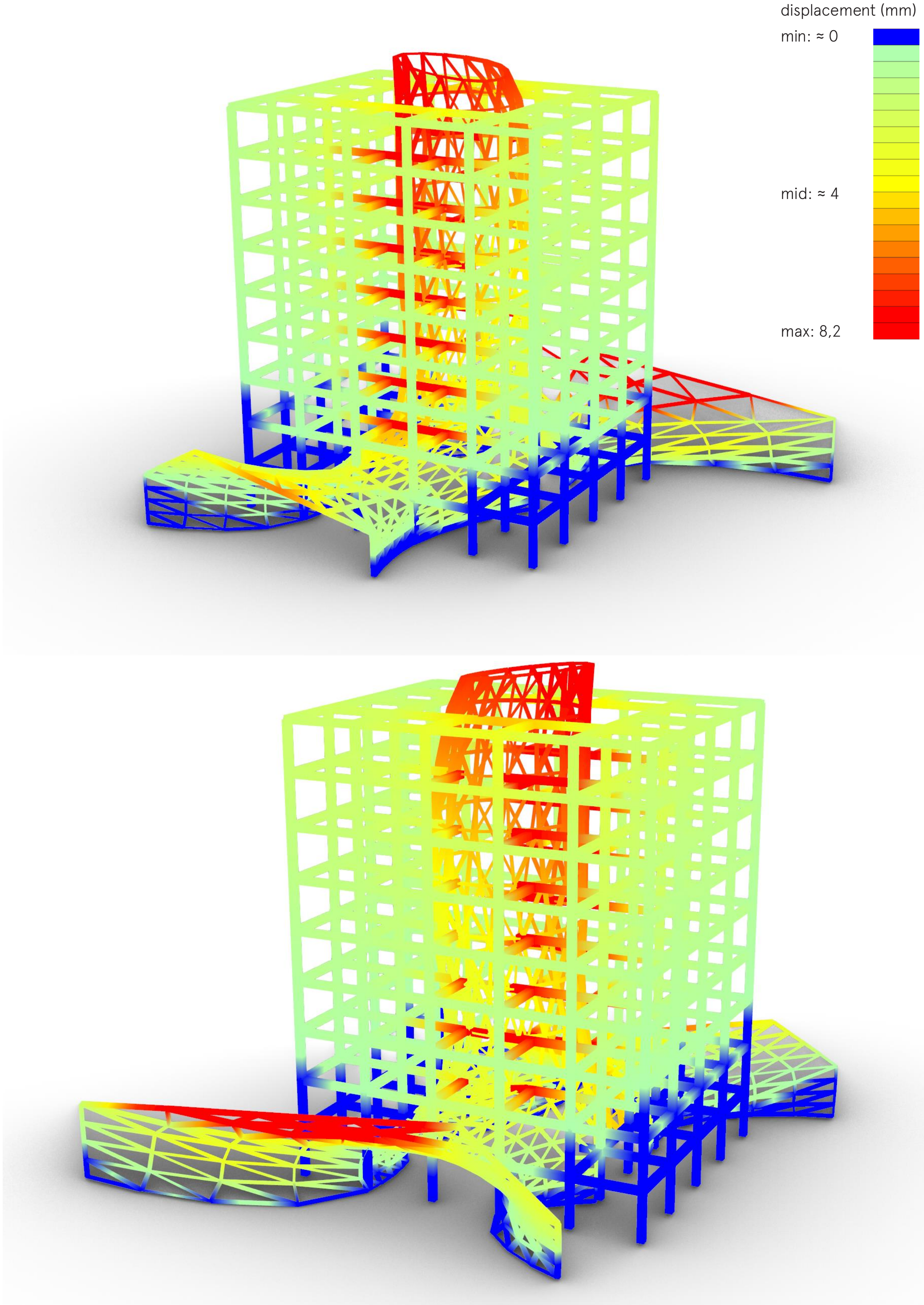


*Fig. 26 FEM model of displacement. Entire structure.
The most displacement of the grid appears in the central part,
connected to the atrium.*

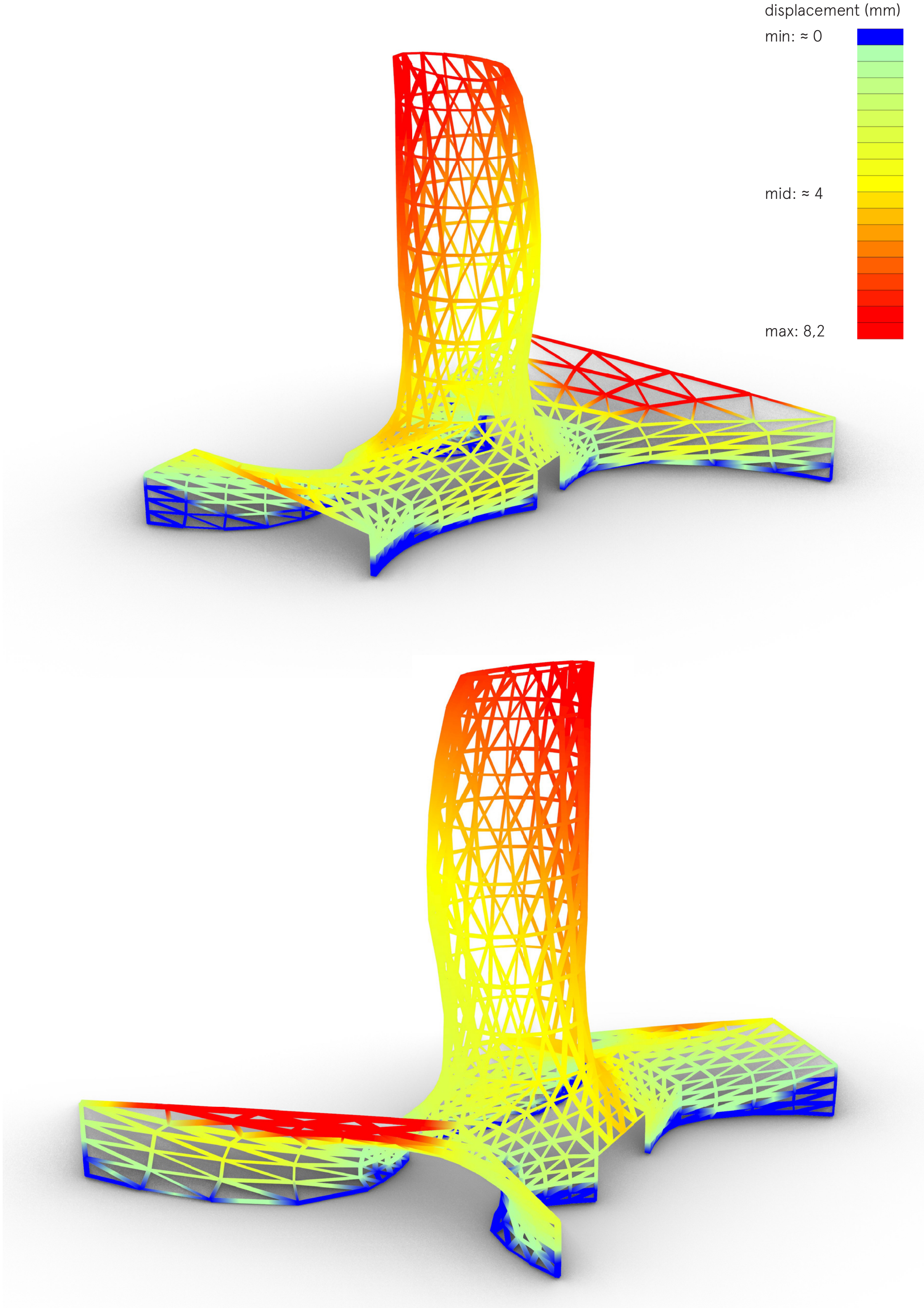


Fig. 27 FEM model of displacement. Freeform structure.
The most displacements appear on top of the entrances' canopies
and at the top edge of the tower.

2_5_ final design

After having all the main elements of the project written with algorithms, the design could be developed and finalised to get a complete working building. The proposal has been submitted under the name #CreativeFarm and has been presented like such:

_ #CreativeFarm

Is it a Building? A Machine? A Village?

The new Koy Bock's Tower of Innovation is conceived as a place for curiosity, exploration and creativity. The design consists of a three dimensional extension of the village in itself, inviting people to go through different atmospheres and paths. The three main functions of the building, a.k.a the working spaces, the auditorium and the lab, are identified by clear volumes. Their positon adapts to the surroundings as the facade generates a variation of views and shadows.

In the center of the building, a freeform structure connects all the volumes together and creates a dynamic experience that contrasts with the rational outer appearence. By extending the exisiting village's axis and river's flow, the 3D surface is literally making the street pass through the building. The node in the center is thus a new inside public space available for exhibition or any other uses.

On the outside, the roofs are made accessible by a promenade going all over the building. Those new outside spaces provide new areas benefitting the whole village. They act like small piazzas elevated in the sky and can be used for various means such as meetings or sports activities. The grid operates as both an unifier and a support. It unifies the different elements of the building by making it a whole and can serve as a support for the future users' creations and technologies.

Fig. 27 View of the lobby from the north entrance.

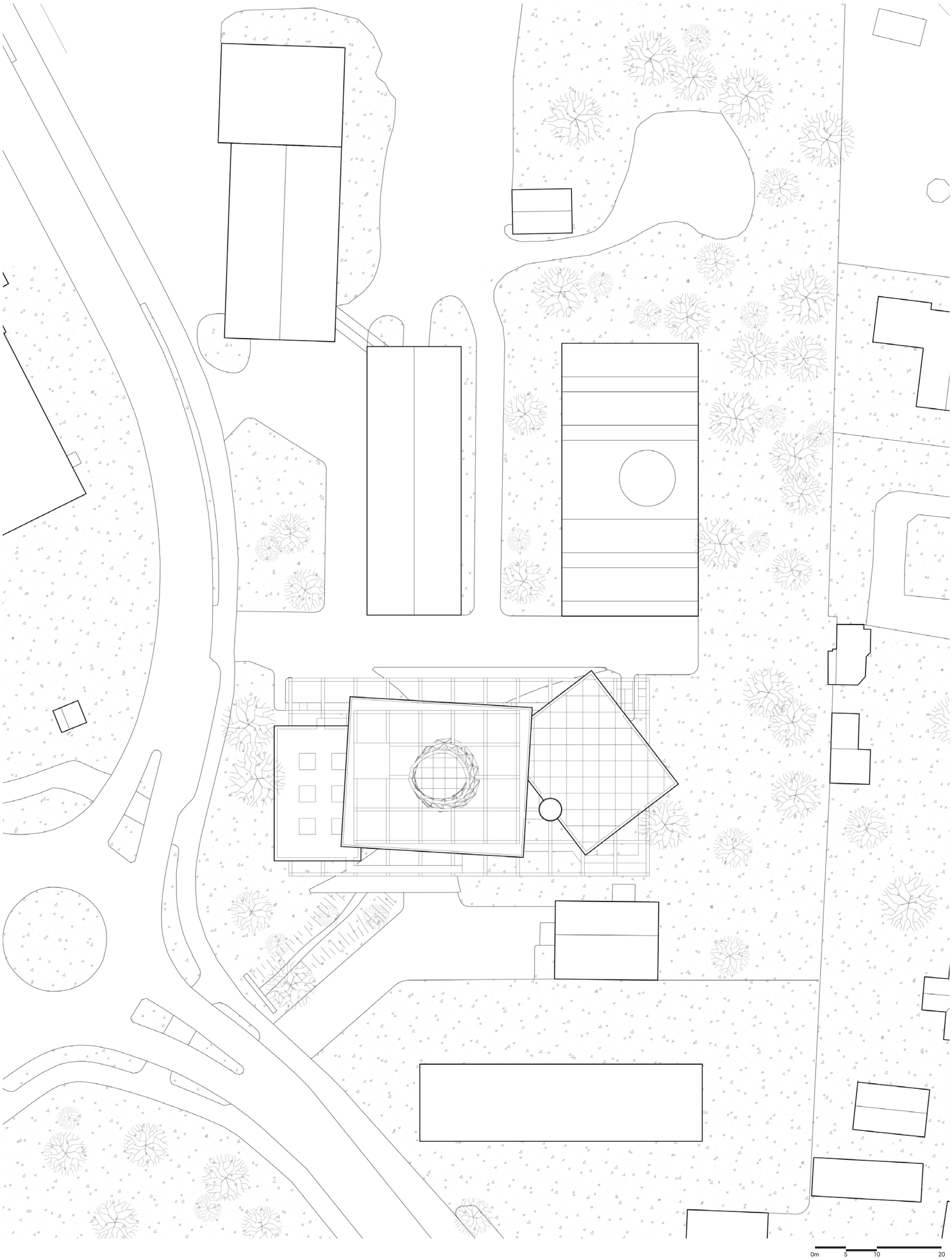


Fig. 28 Site plan with surroundings.

Fig. 29 Birdview of the Koy Bock's Corner Village.

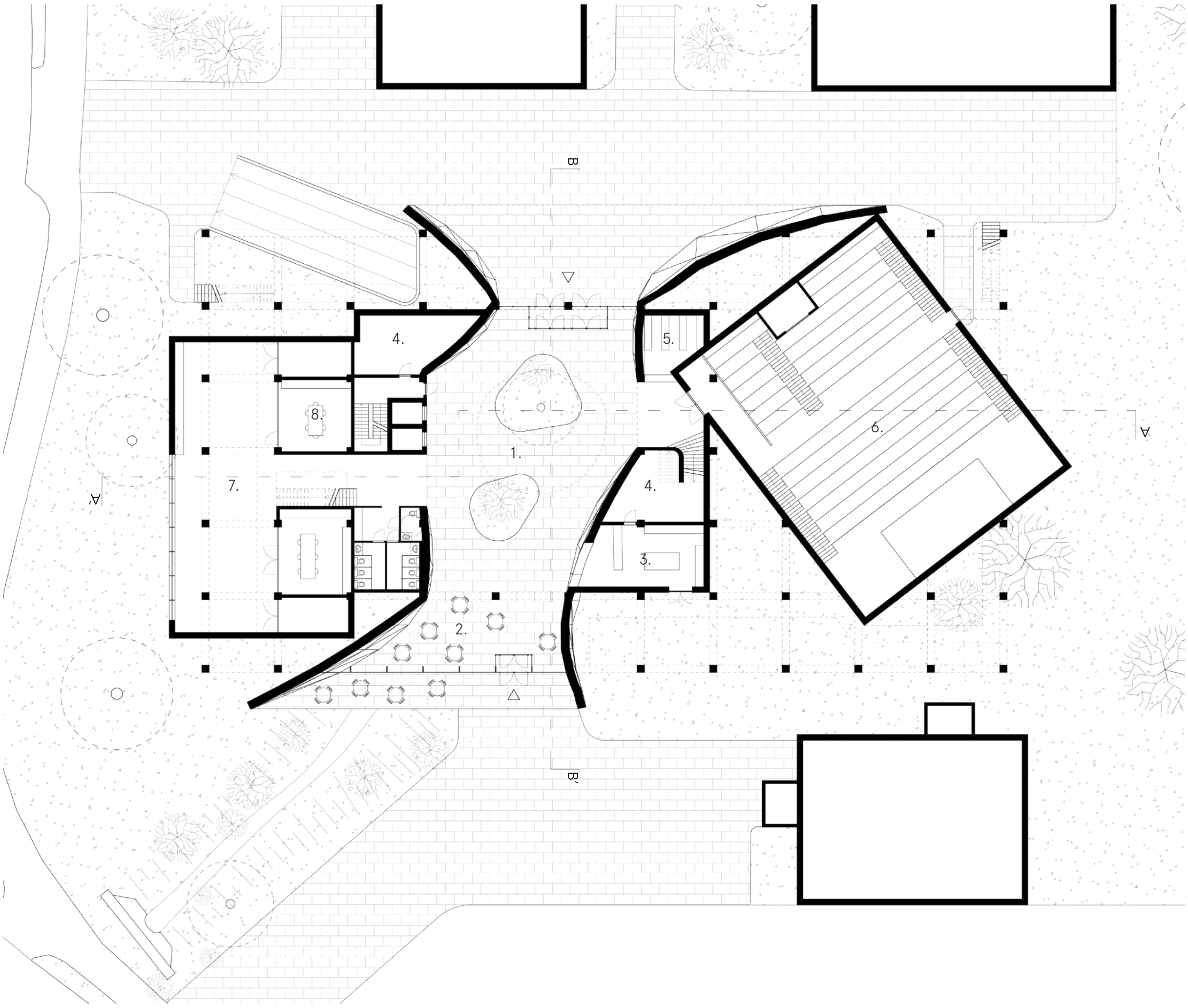


Fig. 30 Ground floor plan.

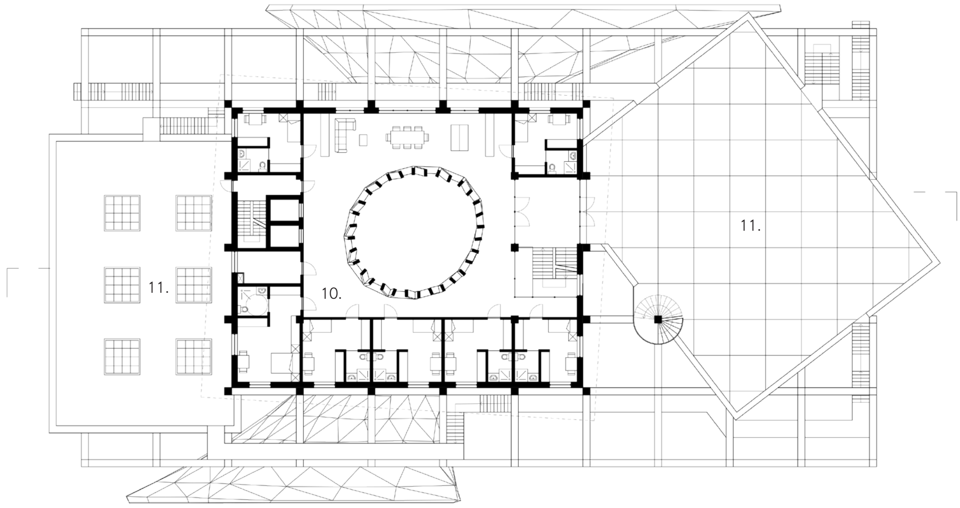


Fig. 32 Second floor.

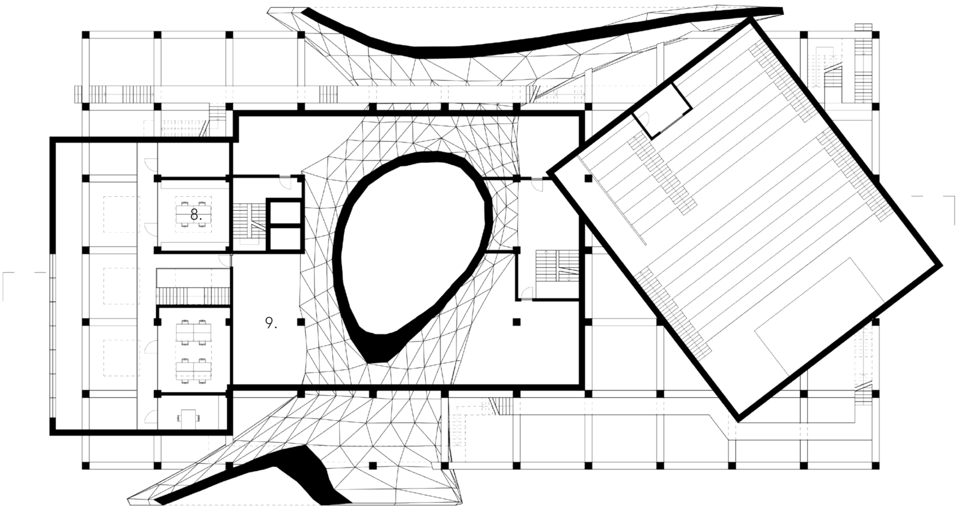


Fig. 31 First floor.

1. Lobby
2. Cafeteria
3. Kitchen
4. Storages
5. Cloakroom
6. Auditorium
7. Fablab space
8. Laboratory rooms
9. Ventilation
10. Researchers lodgings
11. Rooftop piazzas
12. Flexible workspaces

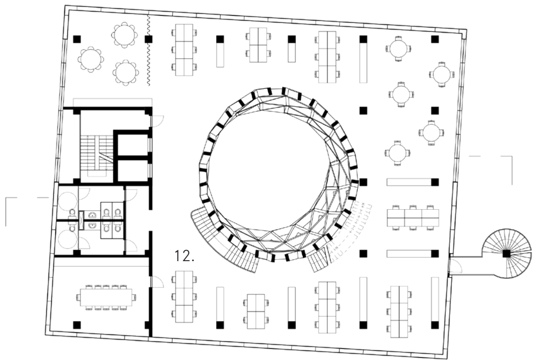


Fig. 35 Fifth floor.

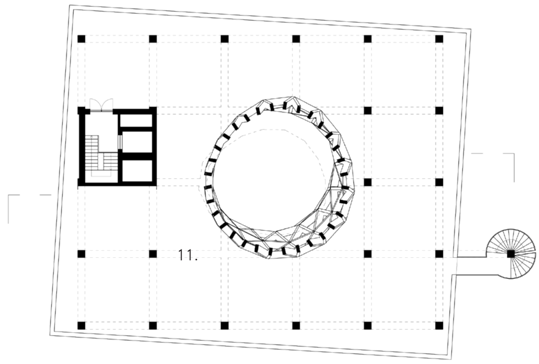


Fig. 38 Rooftop.

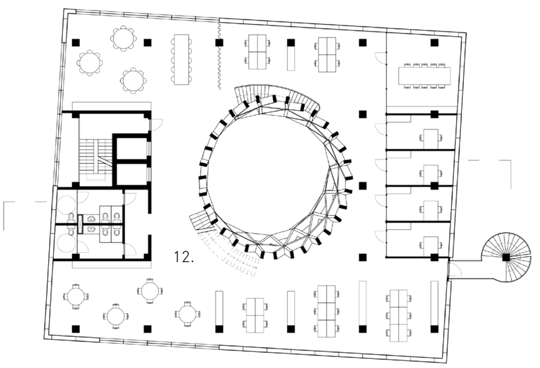


Fig. 34 Fourth floor.

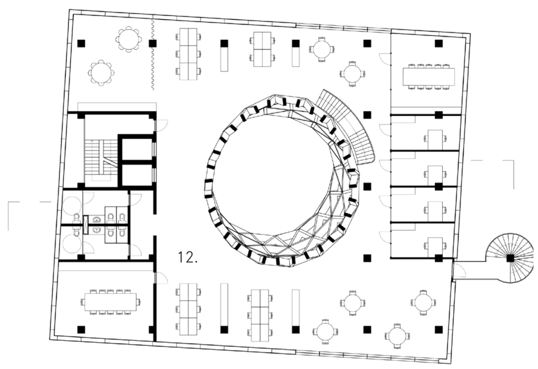


Fig. 37 Seventh floor.

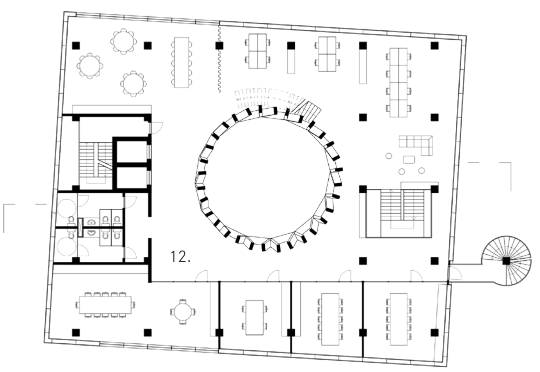


Fig. 33 Third floor.

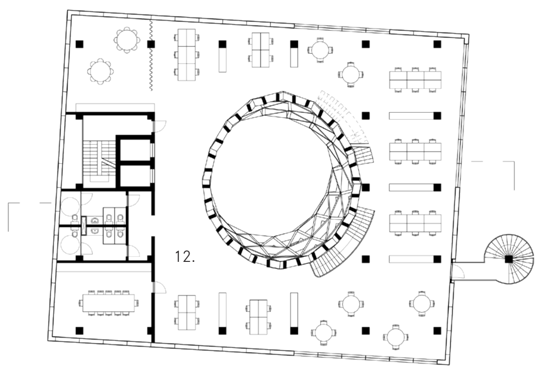


Fig. 36 Sixth floor.

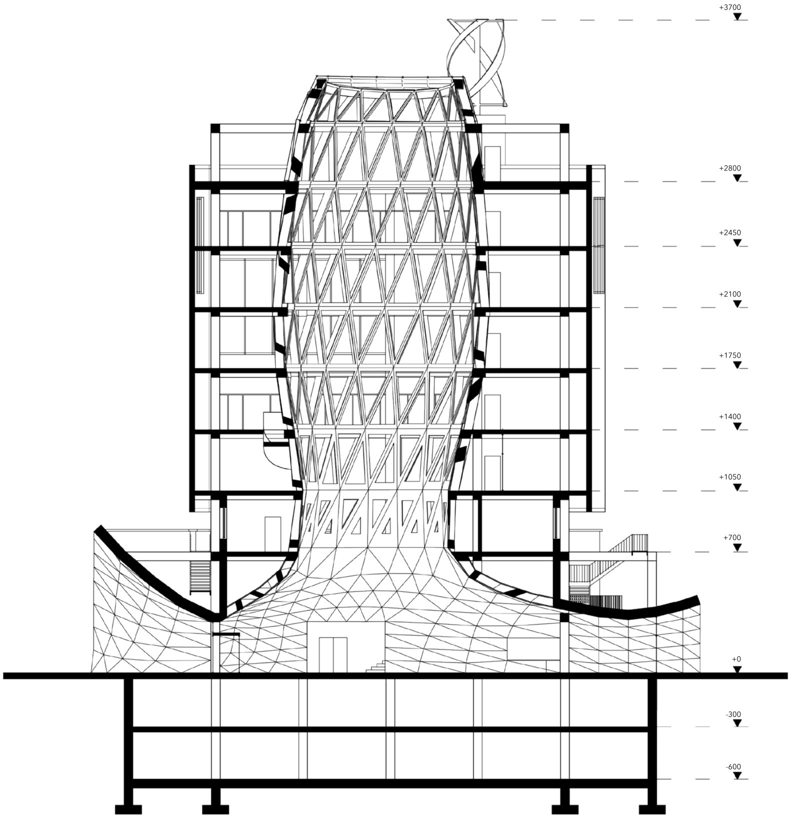


Fig. 39 Section aa'.

Fig. 40 Elevation north west.

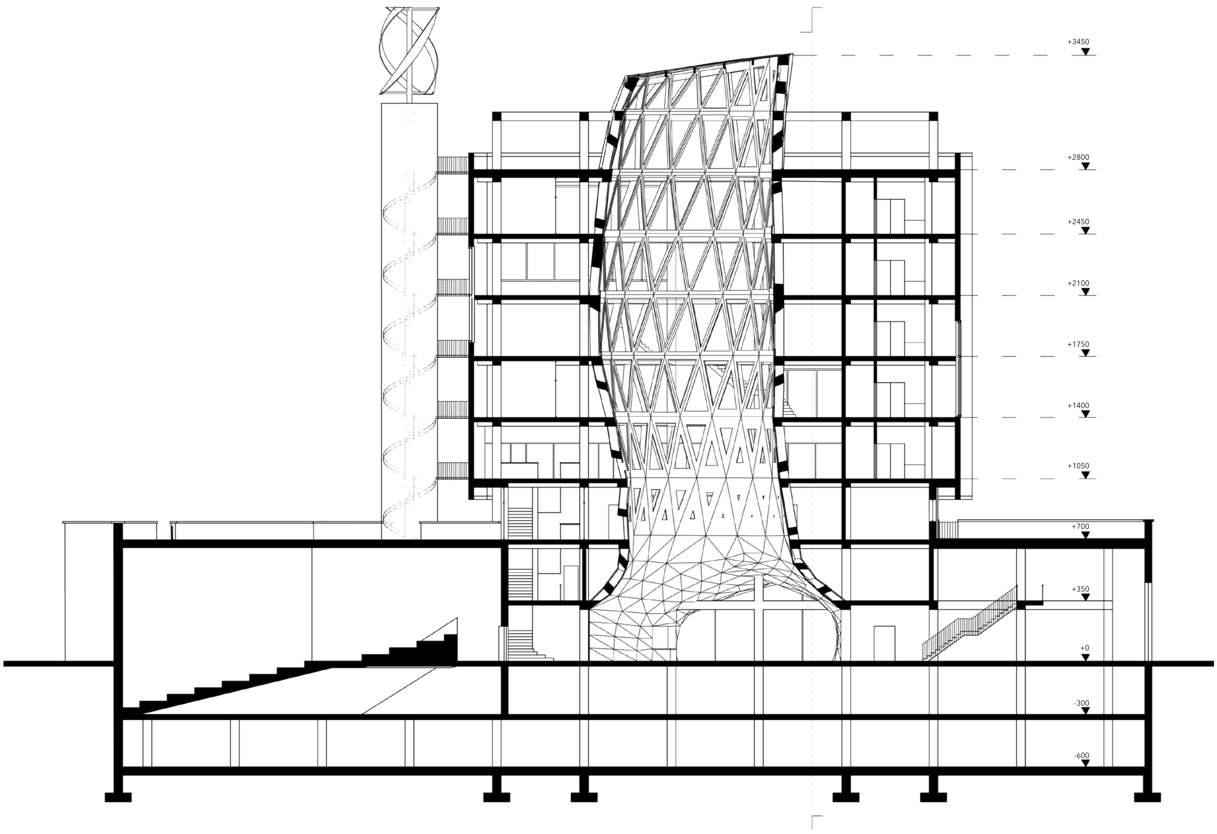


Fig. 41 Section bb´.

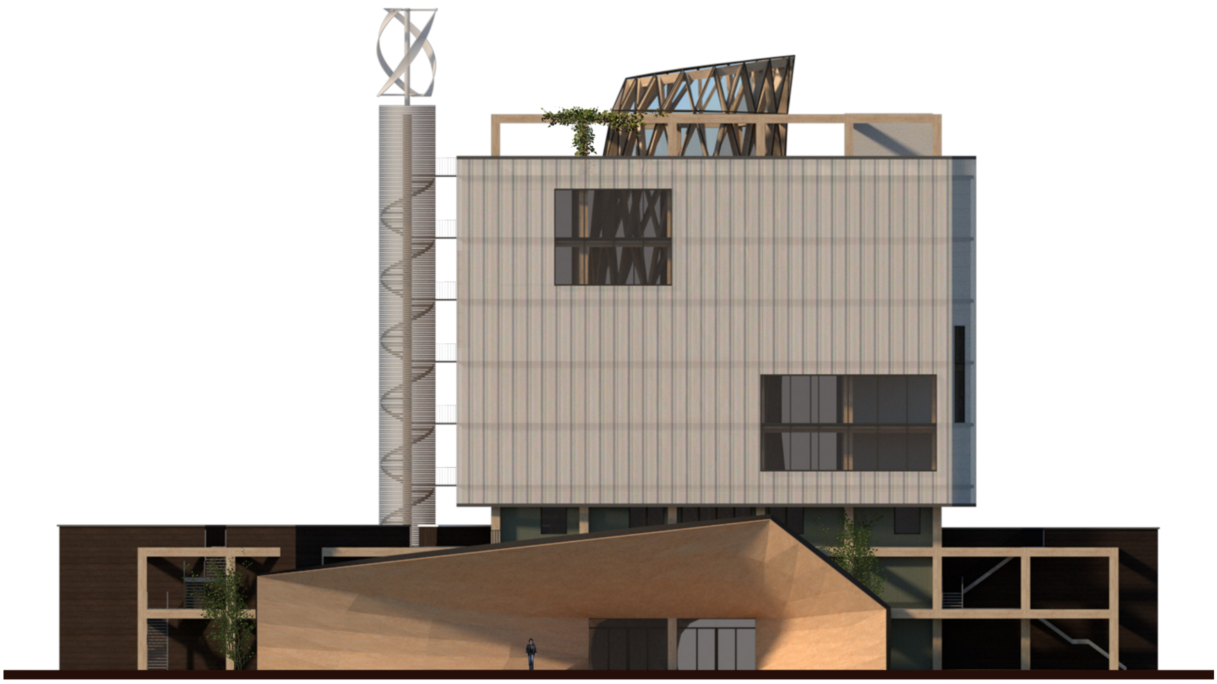


Fig. 42 Elevation north east.

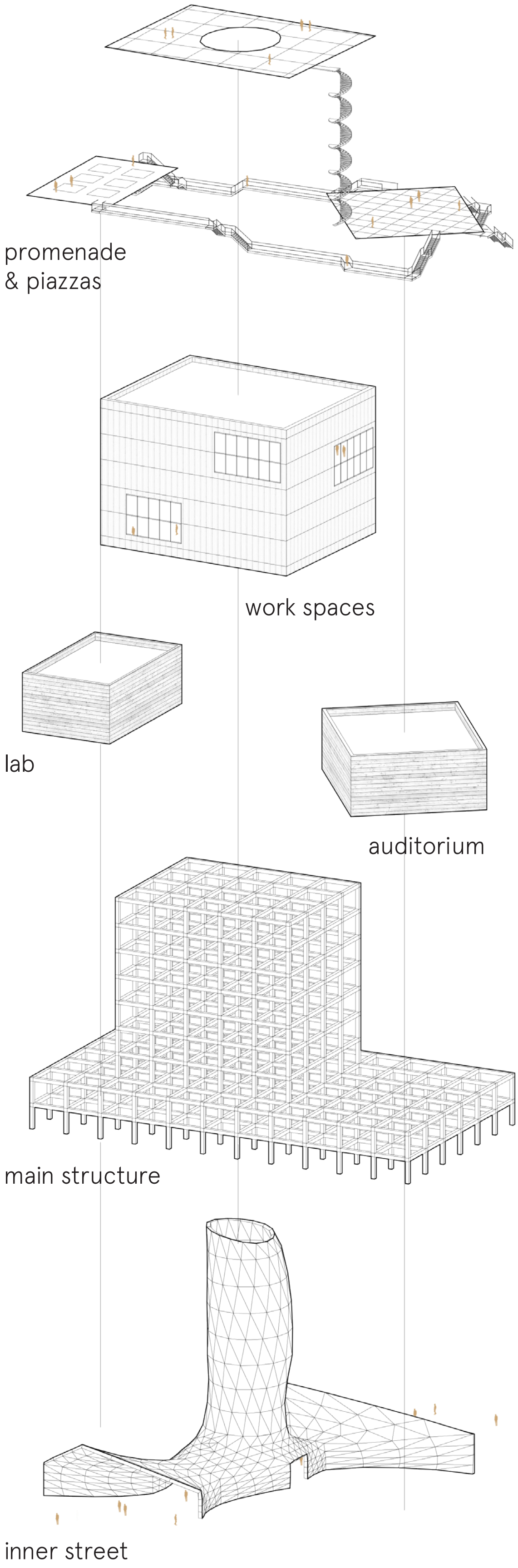


Fig. 43 Split axonometry.

Fig. 44 Night view from Gerbyntie.

2_4_ production phases

Designing with algorithms often implies designing complex details and producing complex blueprints. A freeform structure is most generally linked to the production of a wide number of custom elements and it is thus important to use the good strategy to get a seamless workflow. In this chapter I will try to explore how complex building elements can be produced with the help of automation. The goal is to show how new methods of production can ease the realization of complex structures by bridging the gap between standardization and customization.

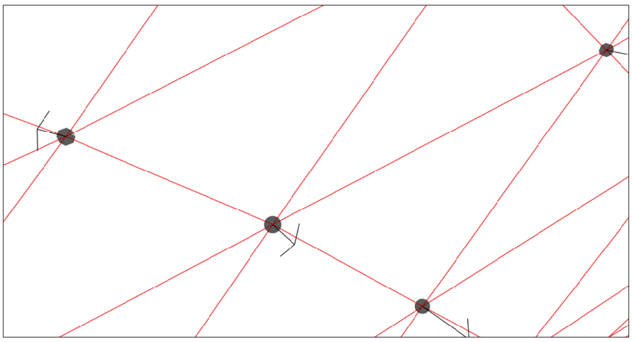
2_4_1_ detailing

This first part will elaborate on the detailing of the three-dimensional shape. Details are fundamentals in architecture as they constitute the realization of the design. They act as the last interface between the conceptual idea of the designer and the physical actualization of it. When designing freeform shapes, the abstraction of the digital model can sometimes be hard to be projected as being a design of an actual construction. As the shapes and objects designed can look vastly different from what we see in the contemporary built environment, it is difficult to imagine how things are made or how they can be assembled. In fact, new forms imply new approaches and details of new genders. They come with a new complexity in the building industry and this must be challenged by the designer who has to integrate custom solutions from the beginning of conception.

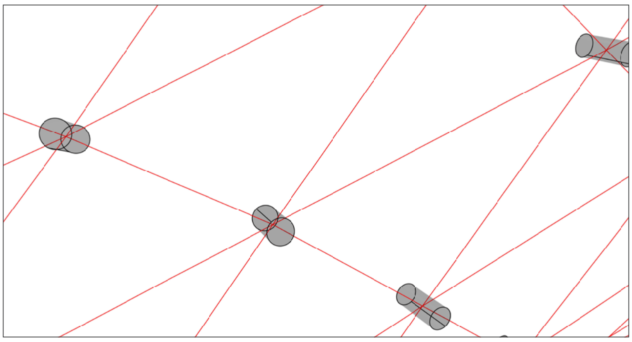
The complexity of this structure resides at the nodes between the glulam elements. The nodes are the linking points of the structural members. They act as rigid connections that hold every piece together. As every element has a different length and orientation, the nodes all have different geometries. Designing such elements would require a large amount of effort with traditional design methods. The total number of joints encountered in this structure rises to 492. It means that the production of such joints requires 492 different details to be designed. A such amount of documents is complex and fastidious to produce and therefore it is necessary to develop an algorithm to automate the process.

Even if the nodes are all different, they follow the same logic: they link the different members. When dealing with computational design, it is often essential to identify what is the main logic behind the design, what are the rules that can lead to the desired final product. In this case, the whole script has been developed on the concept that every joint is a point to which lead lines or axis. The idea consists on first creating a common scheme of the object and then link it to the different situations. The common scheme was made by one cylinder to which flanges are connected. The cylinder would be placed on the node point and oriented perpendicular to the surface. The flanges would be defined by the timbers' axis. They would be generated by first fitting the amount of axis and then adapt their geometries to the axis orientation. This way, the flanges are aligned to the timber elements and all elements' ends can be milled similarly. (fig. 45)

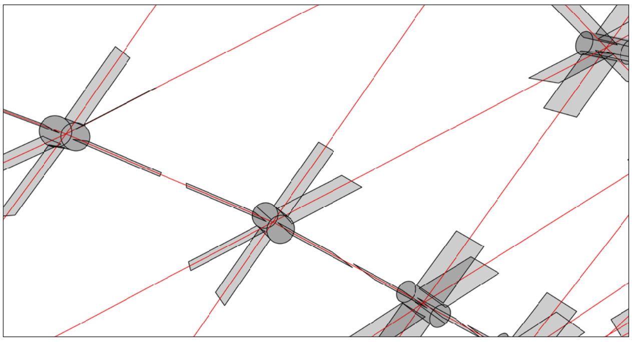
Once designed, the script remains the same whatever the shape would look like. The details are automatically calculated whenever the surface is modified and the designer can therefore freely adapt their shape without worrying about the effort that such changes would imply.



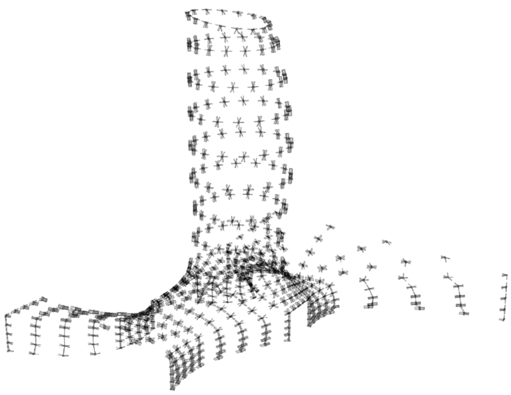
location and orientation of the nodes



extrusion of the cylinders



generation of the flanges



492 similar yet singular joints

Fig. 45 Generation of the nodes.

2_4_2 production

After having all the pieces modelled, the next step is to produce all the documents necessary to produce the pieces. The goal is to get the files that would allow the production company and the builders to manufacture the numerous pieces. The idea mainly consists of extracting the right data to communicate the geometry specifications. While doing that, it is important to create a clear system to label all the pieces in order to keep an efficient workflow between design, production and placement. This process is made using Elefront, a plug-in for Grasshopper that gives the possibility to extract and include data in the digital model.[15]

Here, the 492 joints are defined by the inclinations and number of flanges. The cylinders and flanges dimensions are all the same. The geometry of every joint can thus be fixed by defining the alpha and beta angles of every flange. (fig. 46) A script has been made to extract all the angles and create the necessary documents to produce the steel connectors. (fig. 47 & 48) Every piece has been attributed a label that can be located on a master map showing the position of every elements. (fig. 49)

For the glulam beam, the same scenario has been applied but this time only the length is varying. The overall geometry and the dimensions of the end slots remain the same for every piece.

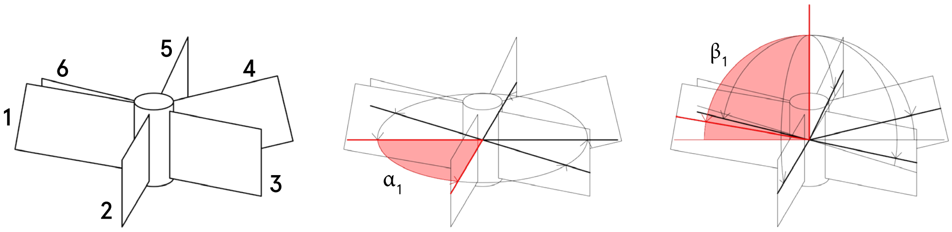


Fig. 46 Explanations of the angles.
Every flange is characterized by two angles.
Alpha defines the polar interval between the flanges.
Beta defines the angle between the cylinder normal and the flange inclination.

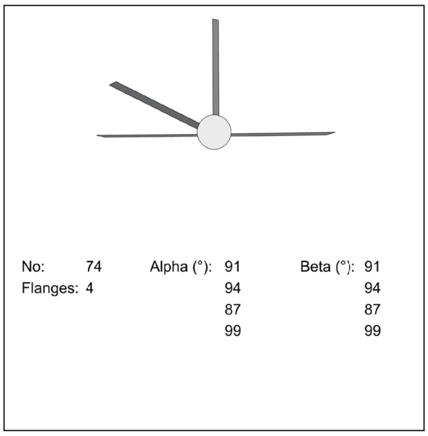


Fig. 47 Extracted characteristics of a node.
Example n°74, can be seen on fig. 48.

15: *Elefront*, © Front, v. 4.0.0, http://www.food4rhino.com/app/elefront

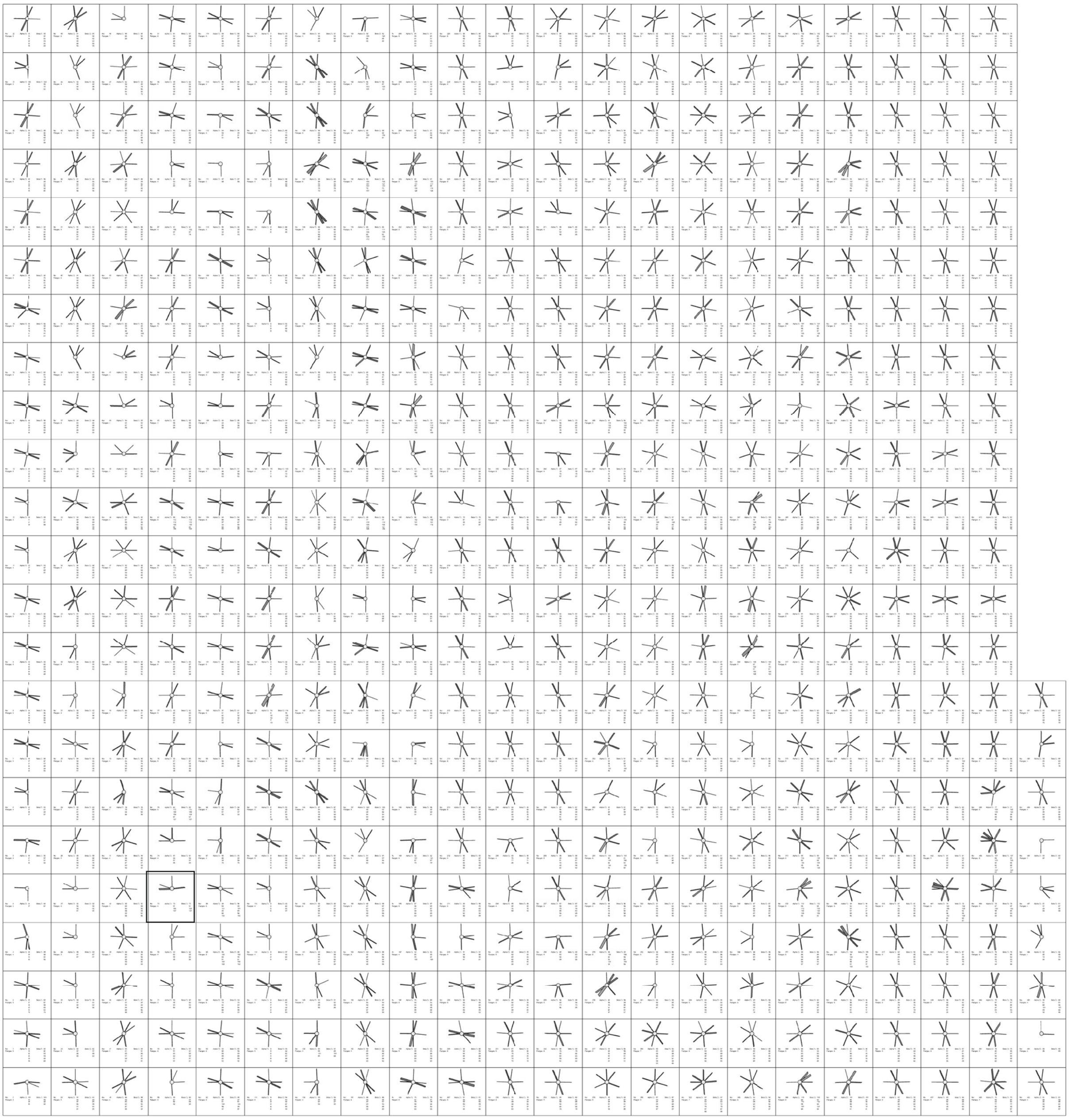


Fig. 48 Series of generated nodes.
All the joints are created with their respective data, including:
node label, number of flanges, alpha and beta angles.

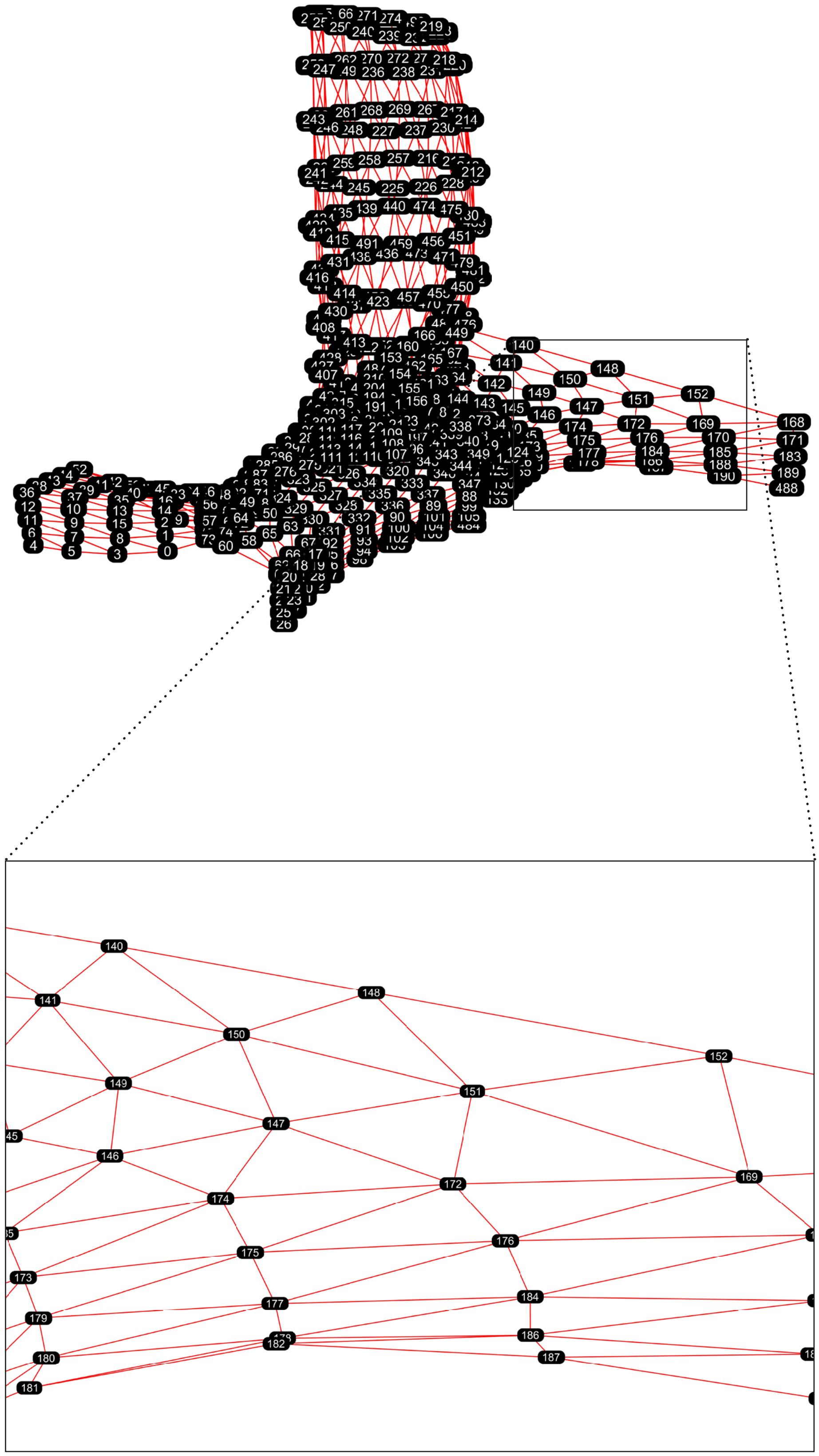


Fig. 49 Master map of the nodes.
Every node is tagged with a number which corresponds to a specific detail.

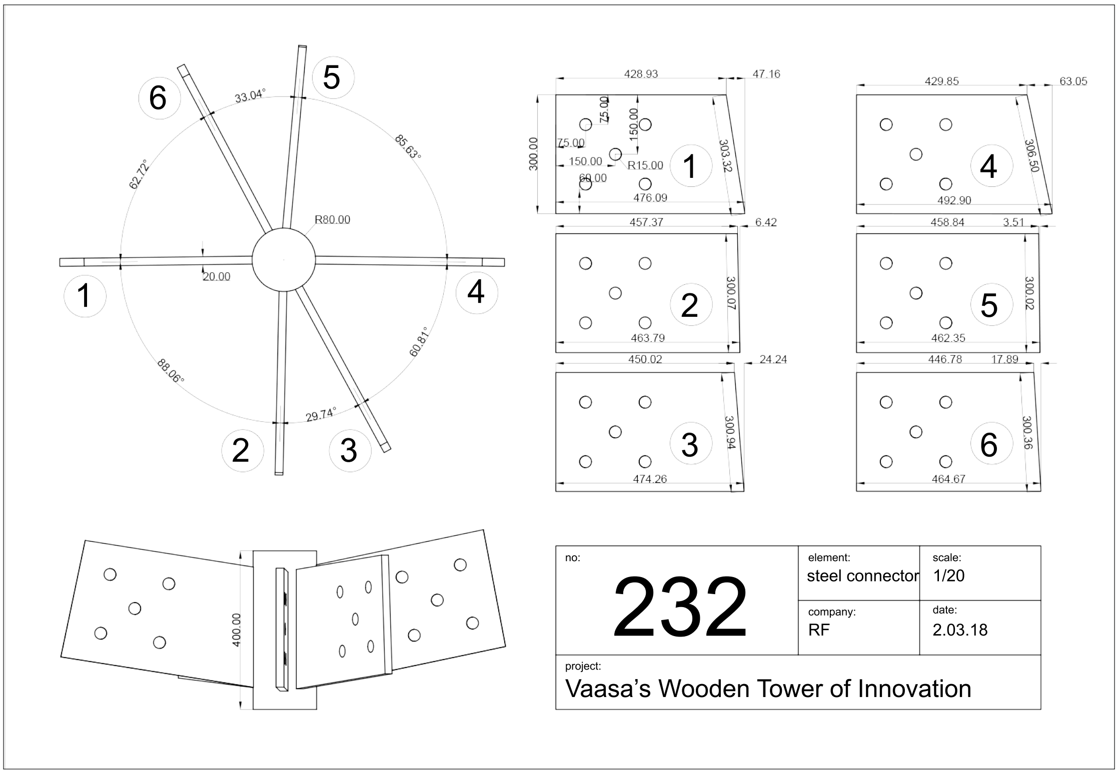


Fig. 50 Detail drawings of a node.

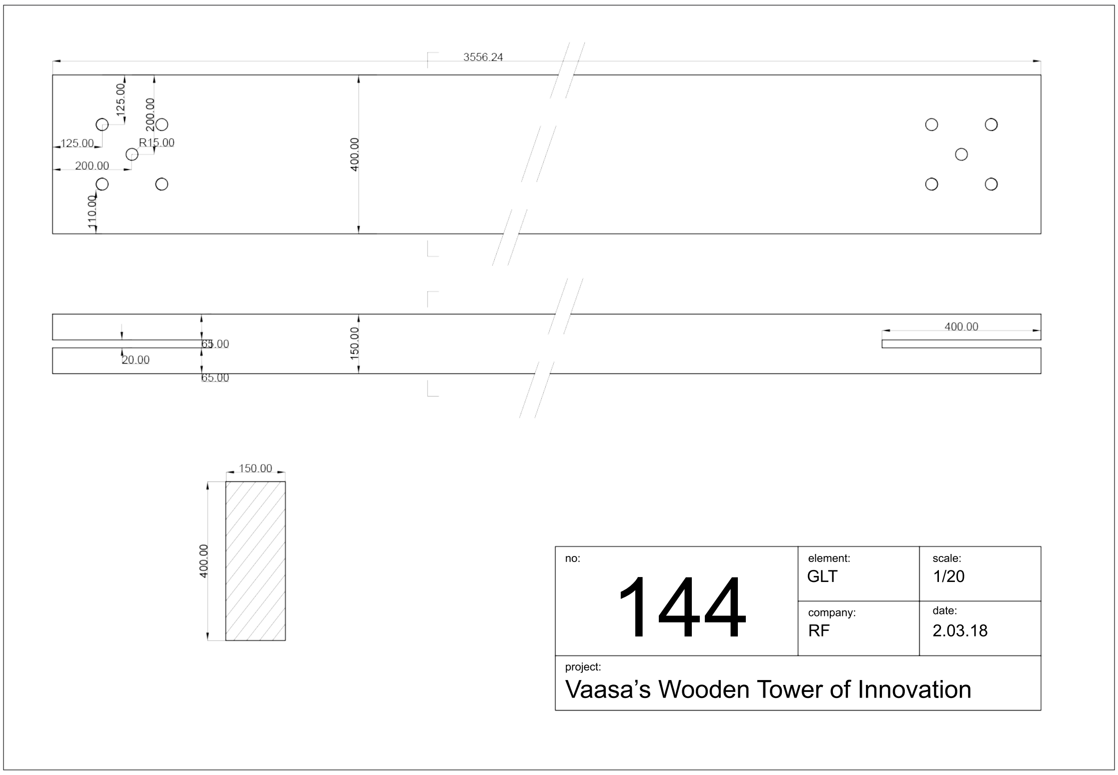


Fig. 51 Detail drawings of a timber element.

3_ Conclusion

outcomes

3_1_ project outcomes

Integrating computation in the design process of an architectural project constitute a real asset. During this project, many elements wouldn't have been made possible without the help of algorithms. The complexity of the design had to be overcome by the creation of custom-made tools. From design to production phases, computation have made the process simple and effective. (fig. 52)

The first and main benefit have been flexibility. If we look at the whole process, we realize that having algorithms implemented in almost every step of the design lets flexibility and progress possible during the entire process. Elements of the project can be developed or analysed without being entirely defined. Thus, the design of the details can be made during the early phases, simultaneously with the shape. Being produced parametrically, the details are automatically adapted while the shape evolves, and both can be developed during the full process. The designer can focus on the changes to resolve and does not need to recreate the design at every adjustment.

Also, as the project deals with complex geometries, a critical aspect consisting of analysing and creating the structure. In this case, computation revealed to be a necessary tool. Keeping a certain integrity while designing is a challenge that has been made possible only by creating appropriate structural tools. The method of FEM has allowed to calculate and visualise the structure with a seamless workflow. Made accessible to designers, this method authorizes the creation of a new type of forms in architecture, with almost no limitation in terms of complexity.

Finally, algorithms have been remarkably useful for the automation of repetitive tasks. Architects often encounter elements that are similar and appear several times in the project. When those elements or the tasks that derive from them can be summarized with rules, it can easily be computed. In this building, the design and production of the pieces forming the freeform surface have been entirely automated. A unique script could generate all the 1864 pieces constituting the surface as well as producing their technical drawings. This has removed a considerable amount of working time in the process and allowed the realization of complex geometries.

All-in-all, we can conclude that algorithms are powerful tools for architects. Depending on the tasks and complexity of the design, architects must find the best way to use computation and create parametric tools that ease and deepen their work, not only aesthetically wise. Algorithms have a wide and versatile potential that is now accessible to designers and now it is up to them to develop their own techniques to produce innovative, appropriate and good architecture.

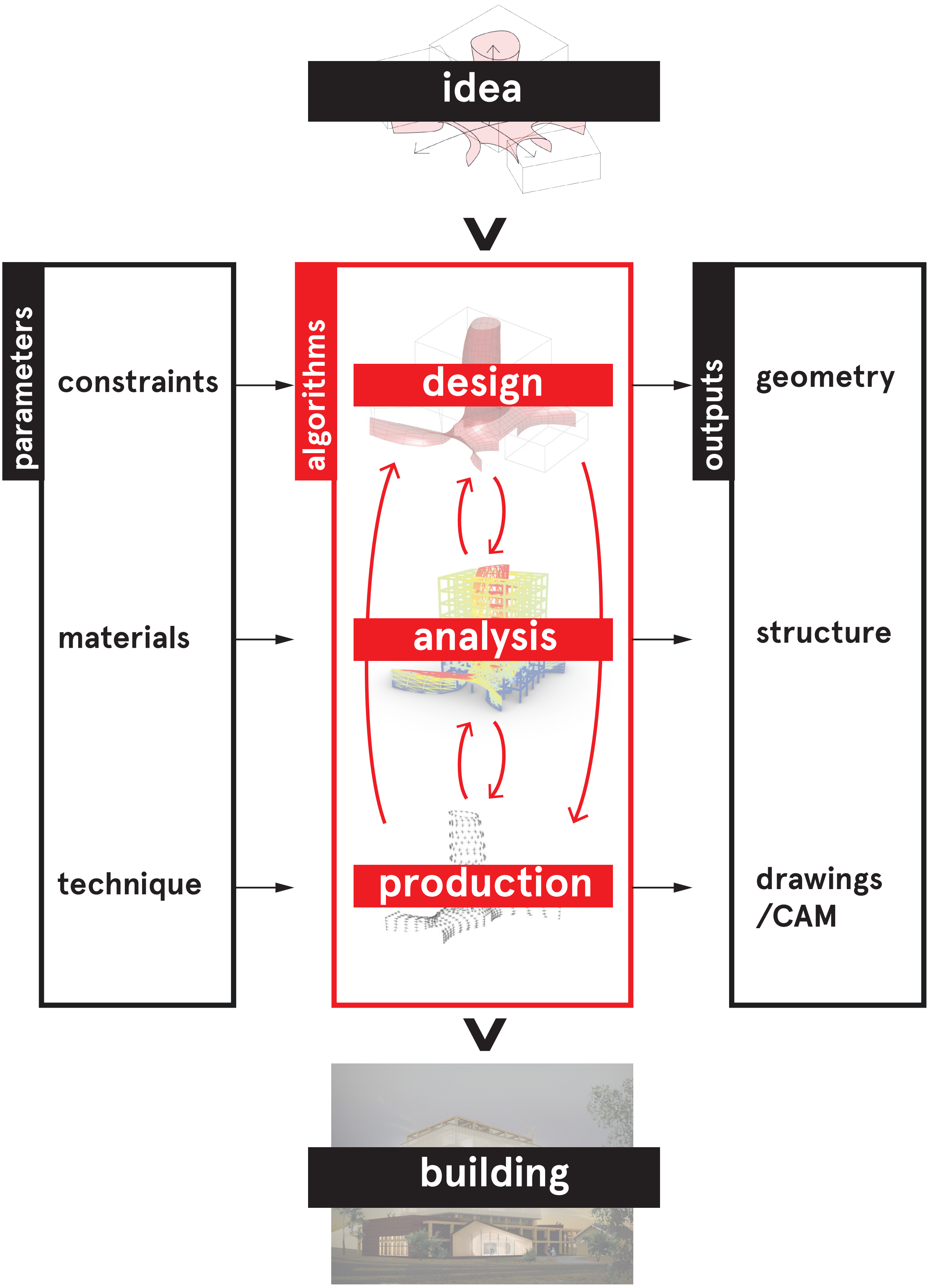


Fig. 52 Worflow synthesis.

3_2_ automation in architecture

As seen in this thesis, algorithms can be integrated in most parts of the design process. The example described here show how computation and automation have been used as tools for the architect who remains the thinker of the project.

Nevertheless, the gap between algorithm as input and algorithm as designer can be considered. As programming has now been made accessible to architects, it also means that elements of the project start to be increasingly automated. As the gap between design and production is on its way to disappear (cf. digital craftmanship), the mechanisms they create can lead to an automation of the design. Even if program, context and materiality remain different for every project and generate the complexity of the architectural practice, the implementation of computational technologies can, to a certain extent, trigger a massive change in the profession and role of the architect. As algorithms can generate designs and drawings, the work of the architect could be resumed as defining the logic of the design and choosing the right parameters to be applied. This extreme scenario is far from being a reality, but as the technology is growing fast and is being always easier to implement, it is essential for designers to understand those new methods and integrate them the most appropriate way.

4_ Bibliography and tools

resources

4_1_ bibliography

_ books & articles

- Carpo, Mario, *The alphabet and the algorithm*, Cambridge: The MIT Press, 2011

- *Eurocode 1: actions on building structures,* Moscow: EU-Russia cooperation on standardisation for construction, 2008

- Gramazio Fabio, Kohler Matthias and Silke Langenberg, *Fabricate 2014*, London: UCL Press, 2014

- Bob Sheil, Menges Achim, Glynn Ruiari and Skavara Marilenna, *Fabricate 2017*, London: UCL Press, 2017

- Schumacher, Patrick, *Parametricism 2.0: Rethinking architecture's agenda for the 21st century*, London: AD Architectural Design, vol. 86, 2016

- Southern, Richard, *The Force Density Method: a Brief Introduction, Bournemouth: Bournemouth University*, 2011

_ websites

- Cambridge Dictionnary, *Algorithm,* Cambridge University Press, 2018, https://dictionary.cambridge.org/fr/dictionnaire/anglais/algorithm, accessed 02.2018

- Encyclopedia of Mathematics,*Minimal surface*, 2017, http://www.encyclopediaofmath.org/index.php?title=Minimal_surface&oldid=28248, accessed 04.2018

- Glued Laminated Timber Association, *Design data*, 2014, http://www.glulam.co.uk/about_designData.htm, accessed 03.2018

- Techopedia, *Computer-Aided Manufccturing (CAM),* 2018, https://www.techopedia.com/definition/4698/computer-aided-manufacturing-cam, accessed 04.2018

_ images

- Fig. 5, p. 23:
Buildipedia, *Zaha Hadid's Heidar Aliyev Cultural Centre: Turning a Vision into Reality,* 2011, http://buildipedia.com/aec-pros/from-the-job-site/zaha-hadids-heydar-aliyev-cultural-centre-turning-a-vision-into-reality, accessed 03.2018

- Fig. 6, p. 23:
Artnet, *The Guggenheim Bilbao, 20 Years Later: How a Museum Transformed a City—and Why the 'Bilbao Effect' Has Been Impossible to Replicate,* 2017, https://news.artnet.com/art-world/the-bilbao-effect-20th-anniversary-1111583, accessed 03.2018

4_2_ tools

_ software

Rhinoceros :
3D modelling software for creation and manipulation of NURBS and mesh geometries.
© Robert McNeel & Associates, v. 5 & 6, https://www.rhino3d.com/

Grasshopper :
Plug-in for *Rhinceros* wich allows algorithmic modelling through a visual programming language interface.
created by David Rutten, v. 1.0.0, http://www.grasshopper3d.com/

_ add-ons for *Grasshopper*

Elefront :
Data management components for *Rhinoceros* objects.
© Front, v. 4.0.0, http://www.food4rhino.com/app/elefront

Kangaroo Physics :
Live physics engine for interactive simulation, form-finding, optimization and constraint solving.
created by Daniel Piker, v. 2.42, http://www.food4rhino.com/app/kangaroo-physics

Karamba :
Interactive, parametric finite element program.
© Karamba3d, v. 1.3.0, http://www.karamba3d.com/

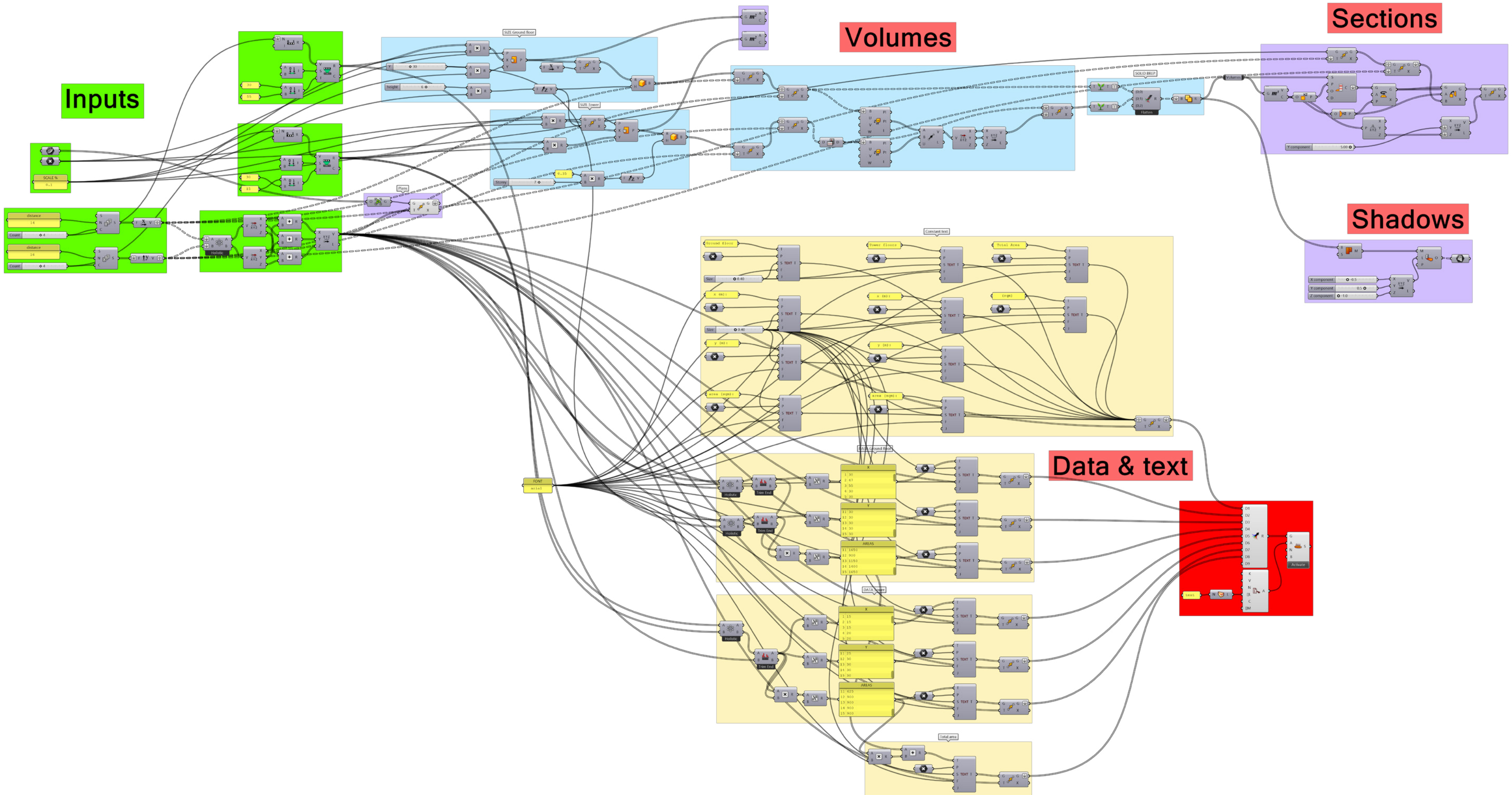
Pufferfish :
Components which focuses on diverse shape changing operations.
created by Michael Pryor, v. 1.7, http://www.food4rhino.com/app/pufferfish

Weaverbird :
Topological modeler that contains subdivision and transformation operators.
created by Giulio Piacentino, v. 0.9.0.1, http://www.giuliopiacentino.com/weaverbird/

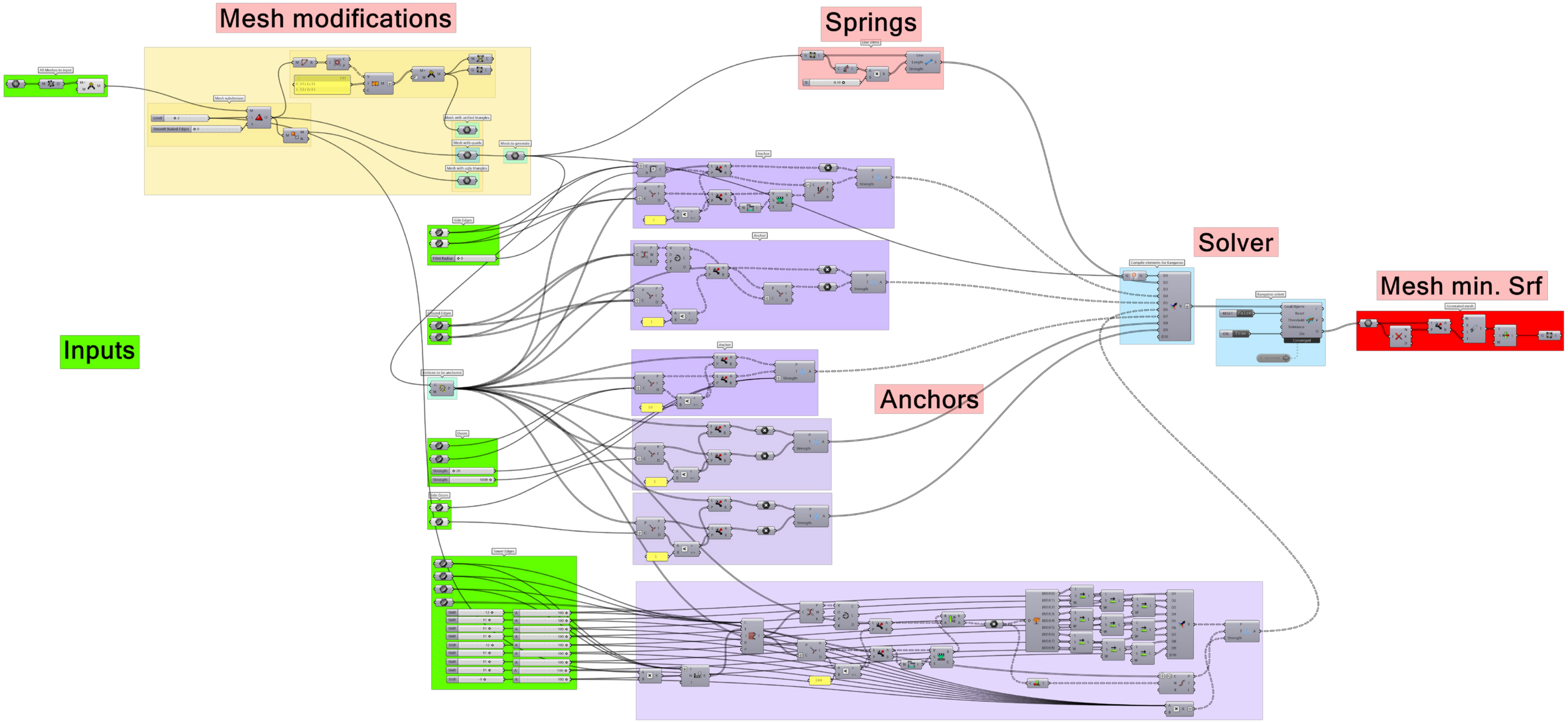
5_ Appendix

algorithms

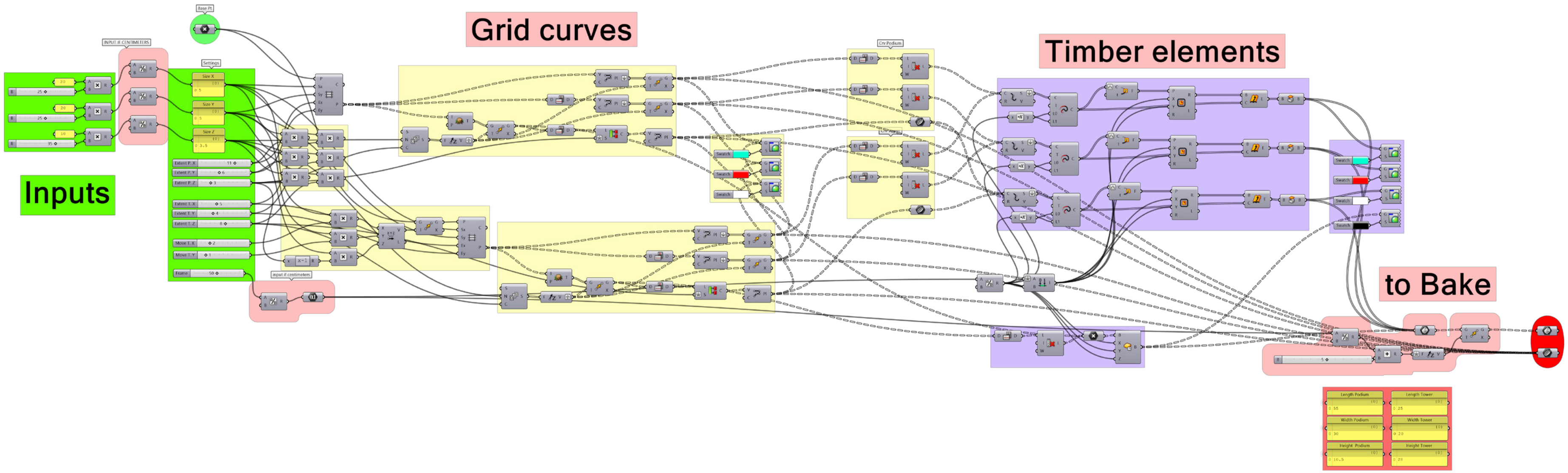
1_ Scale



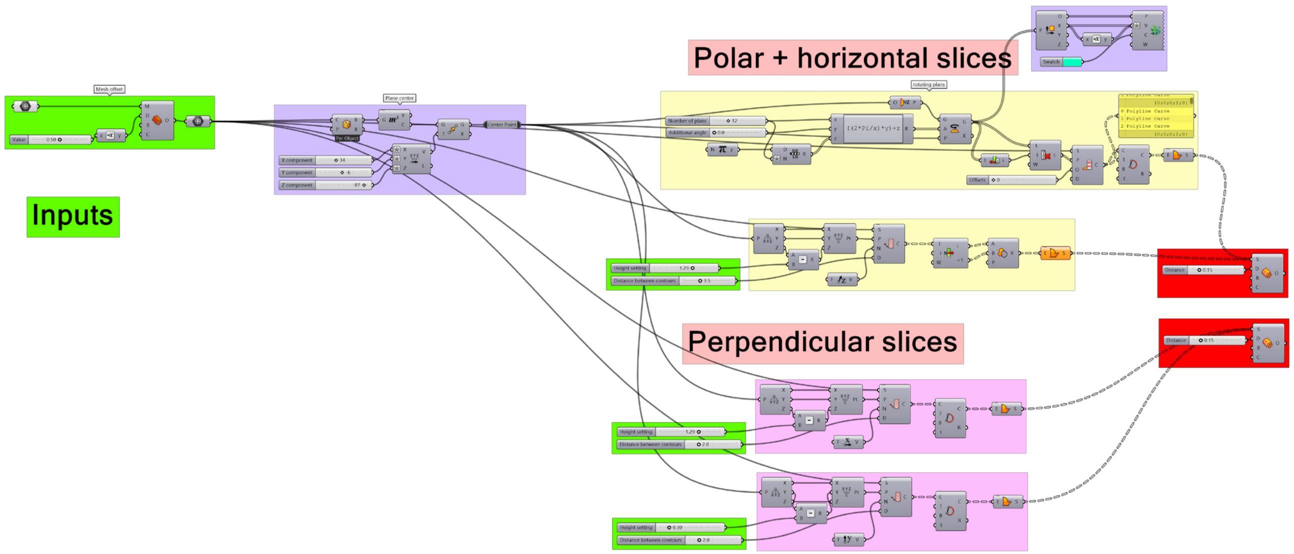
2_ minimal surface and mesh modifications



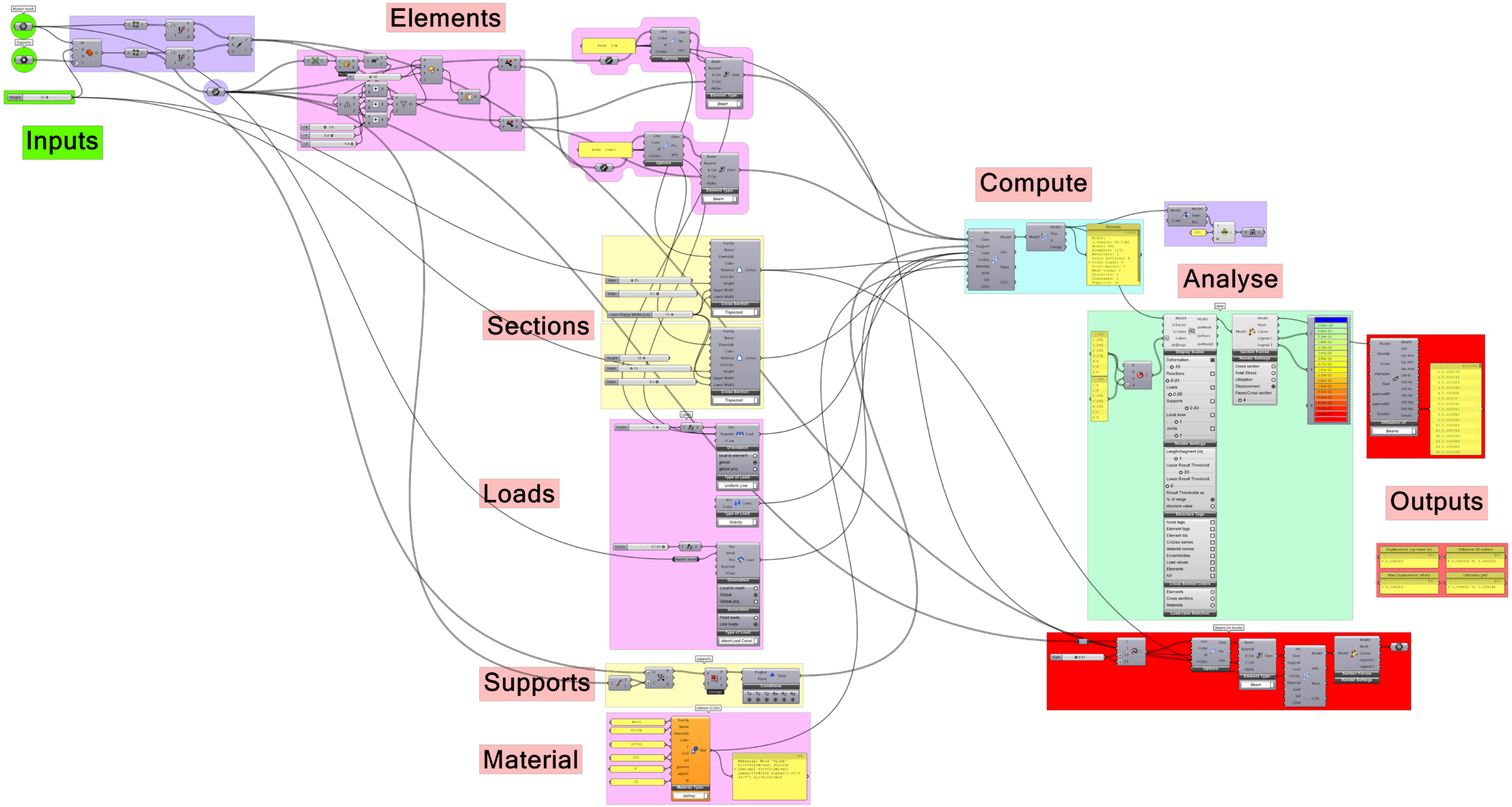
3_ grid



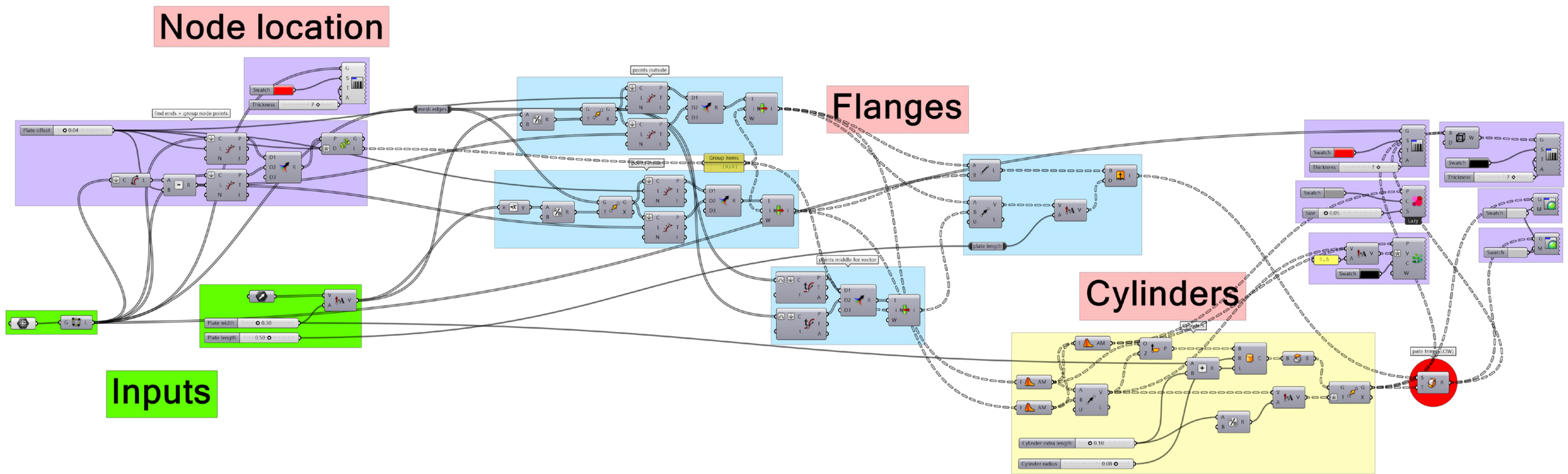
4_ slicing (waffle structure)



5_ FEM model (Karamba)



6_ node generation



7_ production